EMPIRICAL TECHNIQUES TO DETECT ROGUE WIRELESS DEVICES

Bandar Alotaibi

Under the Supervision of Dr. Khaled Elleithy

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

December, 2016

EMPIRICAL TECHNIQUES TO DETECT ROGUE

WIRELESS DEVICES

Bandar Alotaibi

Under the Supervision of Dr. Khaled Elleithy

Approvals

Committee Members

Name	Signature	Date
Dr. Khaled Elleithy		11/21/16
Dr. Hassan Bajwa		11/18/16
Dr. Navarun Gupta		11/18/16
Dr. Xingguo Xiong		11/18/2016
Dr. Mohsen Guizani		11/28/2016

Ph.D. Program Coordinator

Dr. Khaled Elleithy		11/30/2016

Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood		12-5-2016

Dean, School of Engineering

Dr. Tarek M. Sobh		12-5-2016

EMPIRICAL TECHNIQUES TO DETECT

ROGUE WIRELESS DEVICES

EMPIRICAL TECHNIQUES TO DETECT ROGUE WIRELESS DEVICES

ABSTRACT

Media Access Control (MAC) addresses in wireless networks can be trivially spoofed using off-the-shelf devices. We proposed a solution to detect MAC address spoofing in wireless networks using a hard-to-spoof measurement that is correlated to the location of the wireless device, namely the Received Signal Strength (RSS). We developed a passive solution that does not require modification for standards or protocols. The solution was tested in a live test-bed (i.e., a Wireless Local Area Network with the aid of two air-monitors acting as sensors) and achieved 99.77%, 93.16%, and 88.38% accuracy when the attacker is 8–13 m, 4–8 m, and less than 4 m away from the victim device, respectively. We implemented three previous methods on the same test-bed and found that our solution outperforms existing solutions. Our solution is based on an ensemble method known as Random Forests.

We also proposed an anomaly detection solution to deal with situations where it is impossible to cover the whole intended area. The solution is totally passive and unsupervised (using unlabeled data points) to build the profile of the legitimate device. It only requires the training of one location which is the location of the legitimate device (unlike the misuse detection solution that train and simulate the existing of the attacker in every possible spot in the network diameter). The solution was tested in the same test-bed and

yield about 79% overall accuracy.

We build a misuse Wireless Local Area Network Intrusion Detection System (WIDS) and discover some important fields in WLAN MAC-layer frame to differentiate the attackers from the legitimate devices. We tested several machine learning algorithms and found some promising ones to improve the accuracy and computation time on a public dataset. The best performing algorithms that we found are Extra Trees, Random Forests, and Bagging. We then used a majority voting technique to vote on these algorithms. Bagging classifier and our customized voting technique have good results (about 96.25 % and 96.32 % respectively) when tested on all the features. We also used a data mining technique based on Extra Trees ensemble method to find the most important features on AWID public dataset. After selecting the most 20 important features, Extra Trees and our voting technique are the best performing classifiers in term of accuracy (96.31 % and 96.32 % respectively).

ACKNOWLEDGMENTS

My thanks are wholly devoted to God who has helped me all the way to complete this work successfully. I owe a debt of gratitude to my family for understanding and encouragement. I am very grateful to my mother and my father for raising me and encouraging me to achieve my goal. Special thanks go to my wife and my kids; I could never achieved this without their support. I would like also to thank my supervisor Dr. Khaled Elleithy for his valuable suggestions. Without his recommendations this dissertation has not been ever done.

Contents

List of Figures

List of Tables

ABBREVIATIONS

WLAN	Wireless Local Area Network
Wi-Fi	Wireless Fidelity (some resources indicate that it is just Wi-Fi)
DoS	Denial of Service
IP	Internet Protocol
SSID	Service Set Identifier
RTS	Request to Send
ATIM	Announcement Traffic Indication Message
IEEE	Institute of Electrical and Electronics Engineers
DNS	Domain Name System
MITM	Man-in-the-Middle
WPA-PSK	Wi-Fi Protected Access-Pre-Shared Key
WIDS	Wireless Intrusion Detection System
IV	Initialization Vector
EAP	Extensible Authentication Protocol
LEAP	Lightweight Extensible Authentication Protocol
TLS	Transport Layer Security
FAST	Flexible Authentication via Secure Tunneling

PEAP	Protected Extensible Authentication Protocol
VPN	Virtual Private Network
SVM	Support Vector Machine
RSSI	Received Signal Strength Indicator
ISP	Internet Service Provider
CA	Certification Authority
TSF	Timing Synchronization Function
ACK	Acknowledgment
CPU	Central Processing Unit
API	Application Programming Interface
3D	Three-Dimensional
GTK	Group Temporal Key
AP	Access Point
RAP	Rogue Access Point
IDS	Intrusion Detection System
MAC	Media Access Control
BSSID	Basic Service Set Identifier
CTS	Clear to Send
CF	Contention Free
DHCP	Dynamic Host Configuration Protocol
SSL	Secure Sockets Layer
iOS	iPhone Operating System (originally known as iPhone OS, but it can be used for iPad and iPod)

WEP	Wired Equivalent Privacy
WIPS	Wireless Intrusion Prevention System
ICV	Integrity Check Value
RADIUS	Remote Access Dial in User Services
MD5	Message Digest 5
TTLS	Tunneled Transport Layer Security
HTTP	Hypertext Transfer Protocol
DCF	Distributed Coordinated Function
RTT	Round Trip Time
TCP	Transmission Control Protocol
PLCP	Physical Layer Convergence Protocol
IANA	Internet Assigned Numbers Authority
SSH	Secure Shell
IBSS	Independent Basic Service Set
NAT	Network Address Translation
TOFU	Trust on First Use
NIC	Network Interface Controller
PTK	Pairwise Transient Key
PMK	Pairwise Master Key

CHAPTER 1: INTRODUCTION

The usage of wireless networks such as Wireless Sensor Networks (WSNs) and Wireless Local Area Networks (WLANs) have grown in recent years. WSN presents itself as a significant implementation for many applications due to its proficiency to monitor observations and report them to a central unit. Therefore, WSNs have been adopted by several applications such as health monitoring and military surveillance. Additionally, WLANs have gained noticeable attention because of their ease of deployment and the availability of portable devices. Internet usage has moved from stationary computers that are connected to the wired side of the network to mobile devices such as smartphones, laptops, and tablets, which use radio waves to connect to an Access Point (AP) and then to the Internet. People spend a large amount of time online, regardless of where they are. To connect to the Internet, users have to choose between two options. The first is to use a Wi-Fi network, in particular when connecting to the Internet from homes, offices, airports, shopping malls, and universities. The other, more costly option is to use mobile cellular networks. This second option has increased in popularity over the past decade. However, the influence of WLANs remains crucial, especially as Wi-Fi hotspots become ubiquitous. Most wireless users prefer WLANs because, unlike cellular networks, they are free to use [1]. APs are

an integral part of WLANs, providing a coordinated point that manages workstations and connects users to the wired network [2]. Consequently, malicious attacks have increased enormously because of the shared medium that wireless networks use to serve wireless devices [3].

The Media Access Control (MAC) address identifies wireless devices in wireless networks, yet it is susceptible to identity-based attacks. MAC address spoofing is an attack that changes the MAC address of a wireless device that exists in a specific wireless network using off-the-shelf equipment. MAC address spoofing is a serious threat to wireless networks. One of the most common security problems faced by WLANs is the Rogue Access Point (RAP) [4], [5], [6], [7], [8], [9], [10], [11], which is a fake AP that was not installed by the network administrator. As APs have become cheaper, the ability to deploy them maliciously in WLANs has grown tremendously. In the literature, RAPs are classified into four categories: Evil-twin APs, Improperly Configured APs, Unauthorized APs, and Compromised APs [6], [12]. There are also RAP-based DoS attacks that are not classified by the research community. These are deauthentication/disassociation attacks and the forging of the first message in a four-way handshake. It has been estimated that approximately 20% of all APs in enterprise WLANs are in fact RAPs [13], [14], [15].

Some of the early RAP detection methods assumed that the RAP has been inserted by a naive user who wants to access the Internet from, for example, a conference room. Although this was initially true, today it is more likely that the person who has inserted the RAP is a skilled attacker that knows and can evade RAP countermeasures [13]. Current mobile devices contain an array of personal information, such as photos, passwords,

business documents, and important emails. Therefore, connecting to RAPs is highly dangerous, because it could allow attackers to steal sensitive information. Thus, it is vital to secure WLANs and detect suspicious APs.For instance, an attacker can spoof the MAC address of a productive Access Point (AP) in WLAN-infrastructure mode and replace or coexist with that AP to eavesdrop on the wireless traffic or act as a man-in-the-middle (this attack is known as the evil twin attack) [16], [17], [18], [19], [20]. In addition, the attacker can flood the network with numerous requests using random MAC addresses to exhaust the network resources. This attack is known as resource depletion [21], [22], [23].

These threats, along with other existing threats, necessitate the existence of MAC address spoofing detection to eliminate rogue devices. MAC address spoofing detection is very significant because it is the first step to protect against rogue devices in wireless networks. Wireless networks (such as WSNs and WLANs) are integrated into a wide range of critical settings including health care systems such as mhealth applications using machine-to-machine technology [24]. In addition, it is important to detect the presence of the rogue devices in wireless networks to protect smart grids systems such as heating, ventilation, and air conditioning (HVAC) systems [25]. The classical way to deal with spoofing is to employ authentication methods. Although authentication causes overhead and power consumption for wireless devices, it is even more costly to apply authentication to wireless devices that have limited resources. For instance, before authentication takes place (i.e., before establishing the session keys to authenticate frames in a WLAN) the only identifier for a given wireless device is the MAC address. Thus, two devices in the same network that have the same MAC address are treated as legitimate clients, even though one of them has cloned the MAC address of the other.

External solutions [26], [27], [28] that do not require modification to standards and protocols (such as IDSs) have gained attention for decades because of the immediate response to threats and the possibility of eliminating intruders [29]. Some of the IDSs are based on predetermined signatures of familiar attacks, which are saved on the database. The monitored frames are compared with the predetermined signatures. If the match is found, the notification takes place immediately. On the other hand, data mining or machine learning IDSs have an advantage because they do not require predefined static signatures of known attacks. Thus, it can be done automatically through classification or clustering algorithms. There are a wide range of security measurements (such as encryption mechanisms, authentication methods, and access control techniques), but many intrusions remain undetected. Thus, there is a demand to automate the monitoring of WLAN activities to detect intrusions.

There are two known Intrusion Detection methods: anomaly detection and misuse detection. Anomaly detection observes attacks if there is a deviation from the normal behavior by the devices that generate these attacks. Misuse detection recognizes suspicious activities regarding patterns matching of previous built known attacks. Anomaly detection techniques are more likely to detect unknown intrusions and has a high false positive rate. On the other hand, misuse detection techniques have a low false positive rate, but unknown attacks could remain undetected. Several IDSs are considered to be rule-based (in which the system fulfillment depends on security experts who build the rules). Considering the vast amount of WLAN traffic, it is so expensive and slow to build the rules. The rules have to be modified manually and applying new rules is a hard and time-consuming task. To overcome the aforementioned limitations, data mining or machine learning techniques take

place to discover important patterns of large data sets. It can build intrusions patterns which can be used for misuse detection techniques based on classification, and to build profiles for normal behavior (to detect intrusions by anomaly detection techniques).

We propose a solution that is based on the Random Forests ensemble method [30] and a hard-to-spoof metric, namely the Received Signal Strength (RSS). Random Forests-based approaches have been proposed in several applications and systems including Intrusion Detection Systems in the wired networks [31], [32], spam detection [33], and phishing email detection [34]. However, Random Forests has not been used for similar issues as the one that we are solving in this proposal. Our problem depends entirely on the location of the legitimate and the attacker devices. The important feature that we utilize is the RSS that belongs to the physical layer. On the other hand, the wired IDSs utilize the upper layers such as Application, Transport, and Network Layers; some important features are service type (i.e., telnet, http, or ftp), the presence of JavaScript, and the number of links in the email. We also propose another solution based one class Support Victor Machines to deal with difficult situations such as unreachable locations. RSS measures the strength of the signal of the received packet at the receiver device. RSS can be affected by several factors such as the transmission power of the sending device, the distance between the sender and receiver, and some environmental elements such as absorption effects and multi-path fading [35]. Normally, the wireless device does not change its transmission power, so the degradation of the signal from the same MAC address suggests the existence of MAC address spoofing [36]. We carried out an experiment in a "Small Office and Home Settings" live test-bed using WLAN devices to evaluate our proposed solution with the help of two air-monitors acting as sensors. The sensors are capable of sniffing the wireless traffic pas-

sively and injecting traffic into the WLAN. We used the sensors to passively capture the wireless traffic and send it to the centralized utility for further analysis. Finally, we propose a new misuse detection framework based on some machine learning algorithms and a voting technique and discover some important frame fields that reveal the excising of the rogue device.

1.1 Research Problem and Scope

An attacker can spoof the MAC address of a given legitimate user to hide his/her identity or to bypass the MAC address control list by masquerading as an authorized user. A more effective attack that the attacker can perform is to deny service on a given wireless network [37].

Deauthentication/disassociation: In the IEEE 802.11i standard, it is necessary to exchange the four-way handshake frames before an association takes place between a wireless device and the AP [38], [39]. Once the station is associated with the AP, a hacker can disturb this association by sending a targeted deauthentication/disassociation frame to either disconnect the AP by spoofing the MAC address of the wireless user or disconnect the wireless user by spoofing the MAC address of the AP. A more harmful deauthentication/disassociation attack is to send frames to all of the wireless users using a broadcast address by spoofing the MAC address of the AP [40], [41]. After sending the frame, the AP or the user who receives the frame is disconnected and has to repeat the entire authentication procedure in order to connect again. The attacker can also send spoofed deauthentication frames repeatedly to prevent the wireless user or the AP from maintaining the connection

[42]. There are also other attacks such as the power-saving attack that prevents the AP from queuing the upcoming frames for a given station by requesting these frames for a hacker instead of a legitimate station.

In general, there are three broad groups of attacks that target WLAN users which are flooding attacks, impersonation attacks, and injection attacks. Figure 1.1 shows and example of each group of attacks that explains the threats that WLAN users exposed to.

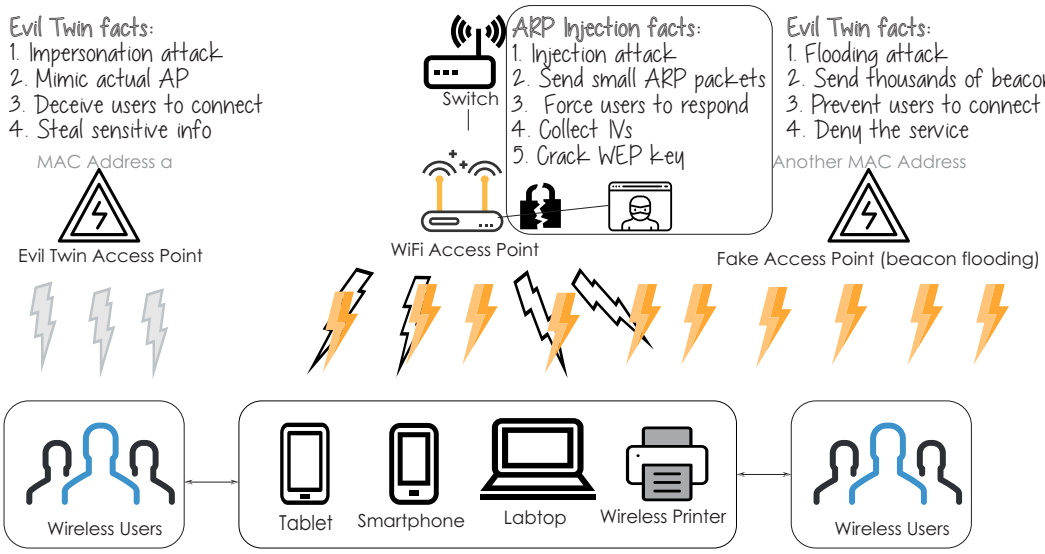


Figure 1.1: Attack scenarios for each group of attacks.

A complete list of flooding, impersonation, and injection attacks and their description are shown in Table 1.2.

Table 1.2: WLAN attacks and their classification.

Attack	Classification	Effect	Description
Amok	Flooding	Dis-connectivity	Sending large amount of de-authentication\dissociation frames to deny the service for long period of time
Beacon		Inability to join WLAN	Sending a stream of beacon frames broadcasting non-existing network in order to make the clients unable to join the preferred network
Deauthentication		Dis-connectivity	Forging de-authentication frames due to the lack of management frames protection using one of the connected clients\APs MAC address to deny the service
Disassociation		Dis-connectivity	Similar to de-authentication attack, however the dis-connectivity is shorter since the target returns to associated state from unassociated
CTS		Disturbance	The adversary continuously sends CTS frames into WLAN to force the clients to deny their transmission
Power Saving		Disturbance	The attacker tricks the AP by sending null data frames to deceive into thinking a targeted client is in sleep mode and cannot receive frames
Probe Request		Disturbance	The attacker transmits probe request frames to the AP to force it to respond with probe response frames to stress the resources
RTS		Disturbance	The attacker sends large amount of fake RTS frames having large duration times in order to reserve the medium to force clients to back-off from transmitting
ARP Injection	Injection	Cracking WEP key	Sends ARP packets into WLAN to collect IVs in order to crack WEP key
Chop-Chop		Key-stream retrieval and frame decryption	The attacker chops the last byte of the packet's encrypted part in order to derive the genuine cipher-text
Fragmentation		Key-stream retrieval and frame decryption	The attack at least needs a data packet from the AP to be initiated. The attacker breaks the packet into fragments and sends the fragments to the broadcast address via the AP in order to retrieve the key stream
Cafe Latte	Impersonation	Cracking WEP key without AP help	Helps speed up the process of cracking WEP key by capturing ARP packets from clients, manipulating the packets, and transmitting it to the clients
Evil Twin		Privacy exposure	A fake AP that advertises the same network name as one of the existing networks to deceive users to connect
Hirte		Cracking WEP key without AP help	The attacker sends ARP request because he or she needs either ARP response or IP packet from the user to perform this attack successfully. The attacker then breaks the packet into smaller packets which speed the process of collecting IVs to crack the WEP key

1.2 Motivation behind the Research

Many techniques have been proposed to detect MAC address spoofing as it is a major threat to wireless networks. First, sequence number techniques [43], [44] track the consecutive frames of the genuine wireless device. The sequence number increments by one every time the genuine device sends either data or management frame. Once the detection system finds an unexpected gap between two consecutive frames, the attacker is detected. Second, the Operating System (OS) fingerprinting techniques [42] utilize the fact that some operating system characteristics could differentiate the attacker from the legitimate device when the spoofing occurs. Finally, RSS techniques [16], [17], [36], [45], [46] utilize the location of the legitimate device that should be different from the location of the attacker if they are not in the same location.

However, there are some limitations in the previous work. Sequence number approaches suffer from some drawbacks: one of the main types of MAC layer frames does not have sequence numbers, which is control the frame. Thus, spoofing of control frames is possible. Also, some of the tools used by the hackers provide the capability of eavesdropping and injecting frames that have sequence numbers similar to the frames of the legitimate device. OS fingerprinting techniques have some weaknesses as well. The first weakness is that the only frame type that can be detected by network layer's OS fingerprinting is data frame. The second weakness is that some of the techniques assume that the attacker spoofs the MAC address using Linux-based operating system tools. This assumption could cause some attackers to bypass the intrusion detection system. The attackers can use a

capability that Windows operating system provides to change the MAC address of a given user. Finally, vendor information, capability information, and other similar fingerprinting techniques can be easily spoofed using off-the-shelf devices.

RSS approaches also have some limitations. Some researchers have reported that RSS samples from a given sender follow a Gaussian distribution, whilst other researchers revealed that the distribution is not Gaussian [47] or that it is not rare to notice non-Gaussian distributions of the samples [36]. As [36] reported, we found that it is not rare to find many peaks in the collected RSS samples. This suggests that the detection techniques [16], [17], [36], [45], [46] (based on clustering algorithms) that are closely related to our proposal are not the optimal solutions because these solutions assume that the samples are always Gaussian. Therefore, their solutions generate false alerts or miss some intrusions if the data is not Gaussian distributed. In addition, when the attacker and the victim devices are close to each other, the means/medians of both devices are close to each other, so distinguishing the two devices becomes hard. Furthermore, we discovered that in multiple cases, the distribution of the data from a single device constructs two clusters, so it is hard for the clustering algorithms-based approaches to perform well in these situations. Motivated by these concerns, we utilized a machine learning algorithm that can deal with both data that are Gaussian-distributed and, more importantly, data that are not actually Gaussian-distributed. Thus, in this article, we proposed a detection method based on Random Forests because it can determine the dataset shape in order to obtain better results and the hard-to-spoof measurement (i.e., the RSS).

1.3 Contributions

This research contributions can be summarized as follows:

1. We develop a new passive technique to detect MAC address spoofing based on Random Forests ensemble method.

2. We compare our work with existing techniques empirically in a live test-bed and find that our technique outperforms existing techniques.

3. We also propose an anomaly detection technique to deal with situations where it is hard-to-cover the whole area.

4. We propose a new WLAN misuse Intrusion Detection framework based on majority voting.

5. We apply feature selection technique based on Extra Trees classifier to improve the accuracy and more importantly to expedite the detection time.

CHAPTER 2: 802.11 STANDARD OVERVIEW

This chapter describes the 802.11 wireless standard at the abstract level. As the focal point of this proposal is APs, we briefly explain the infrastructure mode. The frame types in the 802.11 standard fall into three categories: management, control, and data as shown in Table 2.3. Each type contains several sub-types. Management frames allow WLAN devices to initiate and maintain communications. Control frames govern the wireless links, allowing some stations to access the medium while denying access to others. Data frames convey higher-layer data [48].

Table 2.3: WLAN class 1, 2, and 3 frames

	Management	Control	Data
Class 1 Frames	Beacon, Probe Request/Response Authentication, Deauthentication and ATIM	RTC, CTS, ACK CF-END and CF-ACK	Frames with false ToDS or FromDS
Class 2 Frames	Association Request/Response, Disassociation and Reassociation Request/Response		
Class 3 Frames	Deauthentication	PS-Poll	All data frames

2.1 Connection Establishment Process

Connections are established using several management frame sub-types, as shown in Figure 2.2. The first step is network discovery, which starts when the AP advertises its

existence by broadcasting beacon frames to clients in the vicinity. Clients passively listen to the beacon frames or actively send probe requests to identify APs within range. After receiving a probe request, the AP sends a probe response frame that contains important information such as the supported rates and capabilities of the network. The second step involves the exchange of authentication and association messages. Authentication is the procedure of sending the identity of the station to the AP through the authentication request frame. Upon receiving the request, the AP either accepts or rejects the wireless user via an authentication response. In an open authentication environment, no identity checking takes place. The association request is sent by the station to enable the AP to allocate resources to the wireless user and to synchronize with the user's NIC. The association response sent by the AP details the acceptance or rejection of the connection [27]. Subsequently, the AP and wireless user can exchange data. Establishing secure communication requires further steps after the association stage, such as the exchange of four-way handshake messages for mutual authentication in WPA/WPA2-PSK or the provision of credentials to the authentication server (i.e., RADIUS [49]) in the enterprise mode before the four-way handshake exchange [50].

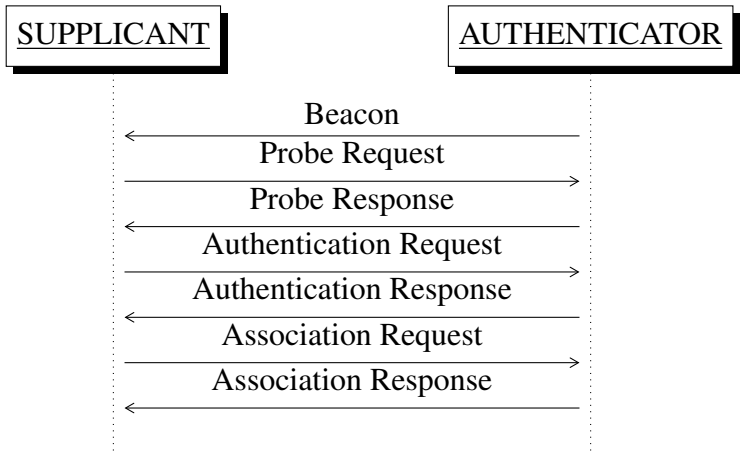


Figure 2.2: Establishing a connection for open authentication

The authentication/association and deauthentication/disassociation state diagram is shown in Figure 2.2. In the first state, the station is neither authenticated nor associated. After the authentication exchange, the station becomes authenticated, but is not associated. Sending a deauthentication message at this stage causes the station to return to the first state, whereas exchanging association frames places the station in the third state, whereby the station is authenticated and associated and can exchange data. Sending a deauthentication frame pushes the station back to the first state, whereas sending a disassociation frame causes the station to return to the second state [37], [51]. To terminate an established connection, the AP disconnects one or all of the connected clients using the broadcast address by sending a deauthentication frame. Both the station and the AP can send a disassociation frame to end the association. For example, the wireless station can send a disassociation frame when the NIC is powering off, allowing the AP to remove the station from the association table and deallocate memory. Deauthentication/disassociation frames are not protected in 802.11i, but are encrypted in 802.11w [52] after the four-way handshake (i.e., exchanging the session keys (PTKs, GTKs)). However, there are some issues regarding the deployment of this standard, namely that millions of devices need to be changed or upgraded. Hence, few WLANs worldwide have implemented this standard. Thus, deauthentication/disassociation DoS attacks remain a problem in WLANs.

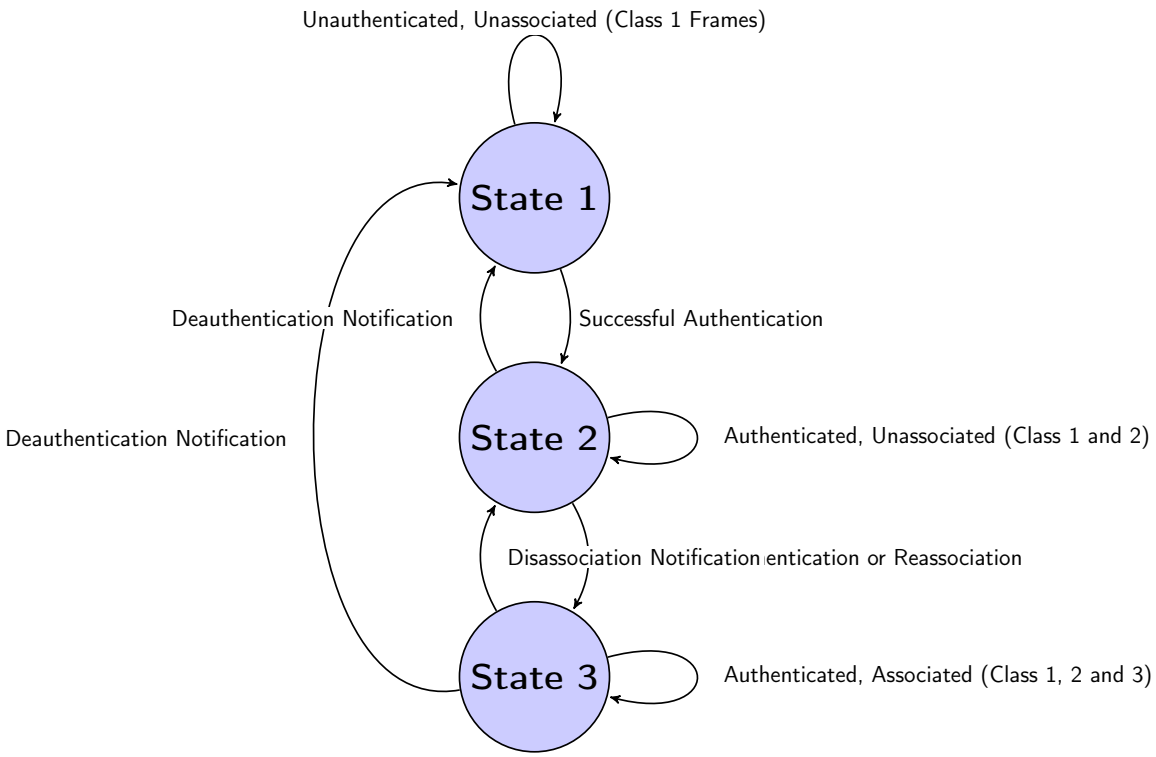


Figure 2.3: Deauthentication and disassociation procedure

2.2 Classification of RAPs

In the literature, RAPs are classified into four categories: Evil-twin, Improperly Configured, Unauthorized, and Compromised. Two more types that can also be classified as DoS attacks are RAP-based deauthentication/disassociation attacks and the forging of the first message in a four-way handshake. These latter two are classified as RAPs in this article, because the deauthentication/disassociation attacks can be sent on behalf of a legitimate AP to disconnect wireless users. This is similar to the Evil-twin attack, because the attacker spoofs the MAC address of the legitimate AP to disconnect associated users. The forged message in a four-way handshake is sent by a hacker who masquerades as the

genuine AP to disturb and block the four-way handshake message exchange between the wireless user and the AP.

2.2.1 Evil-twin

Sometimes referred to as Soft AP or Spoofed AP, we use the term Evil-twin to represent this type of attack. The Evil-twin AP uses a software-based AP installed on a portable device. Thus, a portable device with an external wireless card and a tool such as airbase-ng[1] are sufficient to set up this type of RAP. There are only two identifiers in the IEEE 802.11 standard that can authenticate APs to users. These are the SSID and MAC address (BSSID) of the AP [18]. As these identifiers can easily be spoofed, the AP can be fabricated by an outsider and remain undistinguishable by wireless users. Evil-twin APs come in two forms:

1. *Coexistence* : the legitimate AP and the Evil-twin coexist in the same location. The Evil-twin clones the SSID and MAC address of the legitimate AP [53], and increases its signal strength to force users to connect. It then relays packets through the legitimate AP.

2. *Replacement* : the Evil-twin shuts down the legitimate AP and replaces it. This form of RAP has its own Internet connection.

The first form uses two wireless cards, one built-in to the device and the other a plug-and-play wireless card. The built-in wireless card associates with the legitimate AP, while the

[1] A tool for attacking users and APs.

other wireless card masquerades as the legitimate AP. Packets are then relayed from the Evil-twin's plug-and-play wireless card to the built-in wireless card. The Evil-twin AP is set up by an adversary to listen to users' traffic as they browse the Internet, and to launch several attacks on the victims' devices [4], [19], [54], [55]. The IEEE 802.11 standard states that WLAN clients must connect to the AP that has the strongest signal. To lure users, the Evil-twin can move closer to the users or increase its signal strength to be stronger than the legitimate AP. The Evil-twin then waits for users to connect to it, or may send DoS attacks via deauthentication or disassociation frames on behalf of the legitimate AP to force users to disconnect from the legitimate AP. In practice, an Evil-twin configuration involves more steps to avoid IDSs, such as masquerading AP MAC address and SSID, establishing a DNS server to connect to the Internet, and establishing a DHCP server to automatically assign connected clients with valid IP addresses.

Once a user connects to the Evil-twin, their traffic is exposed to the adversary, who may launch several attacks such as interception, replaying, and traffic manipulation. This can also occur if encryption such as SSL is employed in the user's device. The attacker can act as the Man-in-the-Middle using his AP [18]. To do so, the attacker can easily use tools such as SSLstrip[2] to decrypt the traffic and BurpProxy[3] to generate fake certificates. Because users trust their encryption method, most will accept the faked certificates [56], [57]. Therefore, Evil-twin APs can launch MITM attacks and decrypt encrypted traffic, modify this traffic, and hijack sessions. Evil-twin attacks are very dangerous because of their simplicity. Any mobile operating system such as iOS or Android can be used to

[2]An SSL stripping tool.

[3]An interception tool targeting web applications.

create an Evil-twin. Thus, creating this attack using a smartphone does not necessarily attract attention. Furthermore, easy-to-use tools such as airbase-ng and rfakeap[4] are readily available to help launch the attack.

The second form of Evil-twin attack replaces the legitimate AP, and uses the same Internet connection that the legitimate AP had been using. This type of Evil-twin is harder to detect than the first type, because it clones almost all of the characteristics of the legitimate AP. Additionally, timing approaches that depend on delay cannot detect this type of Evil-twin.

2.2.2 Improperly Configured AP

This type of RAP is not placed by an adversary: it exists in WLANs because the AP is improperly configured. There are numerous situations where the AP can be misconfigured. An administrator who does not have a sufficient security background may choose insufficiently robust authentication or encryption settings. Another example occurs when the AP driver malfunctions or the whole device is worn out. In addition, the AP may become vulnerable after a software update (e.g., firmware with encryption enabled using WPA-PSK or WEP might cause the AP to resume without encryption) [6], [58]. This can open a backdoor to bypass the organization's authentication, allowing unauthorized users to share network resources. This is a hardware-based RAP that is plugged into a switch or router, and there is no malicious intent behind its existence.

[4] A tool that sets up a fake AP.

2.2.3 Unauthorized AP

This type of RAP is installed by an employee or naive user without the network administrator's permission. Although, this AP is not installed by the network administrator, it is considered part of the actual WLAN because it is connected to the wired side of the network, like the legitimate APs. Thus, the unauthorized AP receives and sends wireless traffic from the wireless users to the wired side of the network and vice versa. This RAP can be set up for purposes of convenience, especially in large organizations, to allow employees to gain access to network resources. Unauthorized APs can also be set up maliciously to create vulnerabilities in an organization's security, enabling outsiders to exploit these weaknesses. Thus, unauthorized users who use these RAPs share the medium with authorized users, eavesdrop the authorized users' traffic, and launch attacks against the network resources [6], [58]. This is another hardware-based RAP.

2.2.4 Compromised AP

Security methods such as WPA-PSK and WEP use shared keys to secure the communication between the APs and the wireless users. If an adversary obtains the shared keys used by the APs, the AP becomes rogue [6], [58], allowing hackers to launch attacks and gain access to sensitive information. Hackers with no security background can use simple hacking software; Linux-based operating systems such as BackTrack[5] or Kali[6] provide

[5]Linux-based distribution for ethical hacking.

[6]Another Linux distribution for ethical hacking and security auditing.

multiple tools for hackers to crack the shared keys, such as Aircrack-ng[7].

2.2.5 RAP-Based Deauthentication/Disassociation

This survey focuses on the deauthentication/disassociation attacks that are launched by RAPs to target wireless users. The IEEE 802.11 standard states that deauthentication frames are a notification that cannot be rejected by the receiving wireless client. Thus, the hacker can masquerade as a legitimate AP, and send deauthentication frames on behalf of the AP to the wireless clients to terminate the connection. The attacker can launch a huge number of deauthentication frames to prevent the wireless users from maintaining their connection with the real AP or vice versa. There are three ways that a hacker can launch a deauthentication/disassociation attack:

1. The attacker can create forged deauthentication/disassociation frames on behalf of a connected user, and send the frames to the AP. When the AP receives these frames, it assumes that they were sent by a legitimate user who wants to disconnect from the WLAN. Hence, the AP disconnects the user. This type of attack is beyond the scope of this survey.

2. The attacker can generate forged deauthentication/disassociation frames on behalf of the AP, and send them to a single WLAN user. Once the frame is received, the user disconnects from the WLAN.

3. The attacker can forge deauthentication/disassociation frames on behalf of the AP,

[7] A tool for cracking WEP and WPA-PSK keys.

and send them to all connected users using the broadcast MAC address as a destination address. This attack is severe, because all associated WLAN users are disconnected when they receive the deauthentication/disassociation frame.

2.2.6 Forged First Message in a Four-way Handshake

The purpose of the four-way handshake messages is to verify that the station is in possession of the pre-shared key. For simplicity, we now explain the four-way handshake in WPA2-PSK; this is similar to that in enterprise mode. The PSK in WPA-personal is also known as the PMK. The PTK is derived from PMK, and is installed into the MAC layer [59].

The PTK is split into three keys. The first is known as the Key Confirmation Key (KCK), which is used to verify MIC during the four-way handshake. The other two keys (the Key Encryption Key (KEK) and Temporal Key (TK)) are created after the four-way handshake [27], [60], as shown in Figure 2.3. Before sending the first message, the authenticator generates a nonce (known as ANonce, generated randomly by the AP) and sends it to the supplicant along with its MAC address, known as AA, the sequence number(sn) to prevent replay attacks, and the message number (i.e., in this case msg1). The supplicant generates a random number known as the SNonce, and has the ANonce and the PMK (i.e., entered by the wireless user when choosing the preferred AP from the AP list). Thus, the supplicant can construct the PTK. In the second message, the supplicant sends its own nonce, MAC address, sn, and message number (i.e., msg2) to the authenticator along with the related hash value (i.e., hashed using MIC), which are generated using the PTK that just

has been computed at the supplicant device. The authenticator now has the three important components needed to compute the PTK, namely the ANonce, SNonce, and PMK (i.e., entered initially at the AP captive portal). Prior to sending the third message, the authenticator computes the PTK, verifies MIC, and sends a message including the hash values of ANonce, sn+1, and msg3 along with AA, ANonce, sn+1, and msg3 to the supplicant. The supplicant verifies their receipt by sending a confirmation to the authenticator using the same procedure.

The adversary can mimic the authenticator and transmit a forged first message to the supplicant. This occurs just after the second message has been sent by the supplicant, as the first message is not encrypted (see Figure 2.3). The supplicant then generates a new PTK corresponding to the new nonces that have been generated according to the new received message. Thus, this vulnerability blocks the subsequent handshakes because of inconsistencies in the PTK at the authenticator and the supplicant. Smart attackers can determine the perfect time to send the forged first message by sniffing WLAN traffic, or may simply flood the WLAN with messages, causing a DoS [61], [38].

$\hat{A}$		$\hat{S}$
$Message_1 = (AA, ANonce, SN, Msg1)$		
	$\xrightarrow{Message_1}$	
		$Constucts - PTK$
		$Message_2 = SPA, SNonce, sn, Msg2,$
		$MIC_{PTK}(SNonce, SN, Msg2)$
	$\xleftarrow{Message_2}$	
$Constucts - PTK, verify - MIC$		
$Message_3 = AA, ANonce, IncrementedSN, msg3,$		
$MIC_{PTK}(ANonce, IncrementedSN, Msg3)$		
	$\xrightarrow{Message_3}$	
		$Install - PTK, verify - MIC$
		$Message_4 = SPA, IncrementedSN, Msg4,$
		$MIC_{PTK}(IncrementedSN, Msg4)$
	$\xleftarrow{Message_4}$	
$Install - PTK$		

Figure 2.4: Four-way handshake message exchange

CHAPTER 3: LITERATURE SURVEY

Existing countermeasures can be classified based on whether the technique protects against one or more RAPs, whether the technique is passive or active, and whether it requires protocol modification or special hardware. The following categories are identified to classify the existing countermeasures:

Operator versus Client-side In the operator option, the IDS is implemented on an AP or a router, and the AP tasks are divided between serving the traffic of the wireless users and detecting intrusions. The client-side option focuses on detecting RAPs. There are some challenges to developing a detection system on the client machine, such as:

1. Clients might be limited by the network settings or have fewer privileges than operators.

2. It is difficult for clients to gather WLAN traffic at the network gateway without the operator's assistance.

3. Similarly, it is difficult for clients to have dedicated servers with which to detect RAPs.

Passive versus Active Passive methods simply observe RAPs through wireless traffic, whereas active approaches send test packets to the APs to examine how they react. The biggest problem with detecting RAPs is that they do not reply to active probing. This absence of collaboration has led to passive detection becoming the more popular technique.

Techniques that require special hardware Some techniques require special hardware to perform detection methods, whereas others can simply use smartphones or laptops to perform the task.

Techniques that require protocol modification Some techniques require standards or protocols implemented by the APs to be modified or changed, either by adding more cryptography methods or additional identifiers.

Wireless versus Wired Wireless approaches detect the RAPs using wireless traffic only, whereas wired techniques detect the RAPs by analyzing the wireless traffic that has been relayed by the router/switch at the network backbone on the wired side. Hybrid approaches combine both wired and wireless approaches. Hackers can use various methods to evade the detection methods on the wired side of the network:

1. *The RAP can be hidden behind a legitimate AP*:

   As hotels, airports, universities, and other public WLANs have legitimate APs to which a hacker could connect, the hacker can provide access to friends or outsiders by connecting unauthorized APs to the legitimate AP. Several wired-side detection

methods depend on the usage policy of the switch port; these methods detect the legitimate wireless traffic, and cannot detect an RAP connected to a legitimate AP.

2. *Modifying the pattern of the transmission*:

   Because wired-side detection methods depend on DCF statistics using wireless traffic, hackers can modify their traffic using traffic shaping methods to either add delay or reduce the delay to emulate wired traffic. Thus, an adversary that knows the Ethernet and WLAN speeds can add delay at the application layer to emulate wired-side traffic when the WLAN side is faster than the wired side, and vice versa.

Wireless approaches suffer from expensive sensor deployment. Hybrid techniques are generally good, but hackers can evade the hybrid methods through the wired side.

Techniques that detect all or some RAPs Most techniques focus on Evil-twin detection and indirectly detect RAP-based deauthentication/disassociation attacks. Some techniques detect Unauthorized APs, but the detection of Compromised APs is rare. There is no single technique that detects all RAP types.

The ideal method is one that can detect all RAP types, is passive, does not require protocol modification, and does not require specialized hardware. All existing techniques have one or more of these features, but none of them has all four. In the next two sections, the RAP prevention and detection methods are comprehensively surveyed to identify risks and clarify the restrictions of state-of-the-art detection approaches.

3.1 Available Security Countermeasures

In this section, we explain why available security countermeasures cannot protect against all RAP types. Some countermeasures are designed for WLANs, whereas the rest are adopted from the wired world. This section introduces the most widely used protocols in WLANs to help protect against rogue devices in general, and RAPs specifically.

WEP was developed to encrypt the data transmitted on WLANs. The encryption process in WEP starts by combining the 24-bit IV and the secret key that indicates the encryption/decryption key. In addition, the resulting key is used to produce the key sequence. Furthermore, the plaintext message and the ICV are XORed with the key sequence to produce the cipher text. In the final step, the IV and the cipher text are concatenated. The reverse of the encryption process is the decryption process. There are two characteristic weaknesses with WEP: the IV is frequently reused, and the WEP secret key is not changed often enough. Hence, it is difficult to ensure the existence of two different key streams. Additionally, it is not difficult to attack WEP because it is possible to eavesdrop the IV that is transmitted. Thus, if the sender encrypts two messages using the same IV along with an original message, it is feasible to decrypt the encrypted messages using the XOR operation. The key can then be recovered once the attacker gathers the key streams [62]. Because WEP is not secure, it does not protect against all RAP types.

PSK is used to encrypt wireless traffic between the wireless user and the legitimate AP. One weakness of PSK is that the protocol does not allow any update or renewal property,

so distributing the key in a secure manner is difficult. Some organizations distribute the key on a printed receipt, whereas others use easy-to-guess passwords, so it is easy to intercept the four-way handshake messages and perform a dictionary attack to obtain the key. Thus, network administrators must renew the PSK on the AP manually, and provide the key to all clients that participate in the network. Therefore, this procedure is time consuming and insecure, especially if the administrator chooses an easy-to-guess pass-phrase [63]. This method can protect against Compromised APs and Evil-twins if and only if the network administrator chooses a hard-to-guess password and distributes it in a secure manner.

WPA-Enterprise Mode (802.1x) IEEE 802.1x [64] was designed as an access control method to allow users to connect to the network. It also provides port security to prevent unauthorized access to network resources. IEEE 802.1x has three important components in a given wireless network: the supplicant, i.e., the wireless user that intends to join the wireless network, the authenticator, who is responsible for providing access, and the authentication server, which is responsible for making authentication decisions. IEEE 802.1x uses existing protocols to accomplish its objectives, such as EAP [65], [66] and RADIUS. EAP provides many methods, each having different properties that are suitable for a specific wireless network environment. The system administrator is responsible for choosing which EAP method is used in the wireless network that he/she administrates [67]. EAP uses challenge/response messages. The authenticator is responsible for asking the supplicant to provide more information before deciding which authentication method to use in the link control phase. The EAP authentication process consists of two important elements, requests and type fields. The authentication phase uses either success or failure messages.

There are several EAP methods for different network environments, such as EAP-MD5, LEAP, EAP-TLS, EAP-TTLS, PEAP, and EAP-FAST. One of the most secure is EAP-TLS, which uses public key cryptography to provide certificates to the users. EAP-TLS provides certificates to both the client and the server, and supports mutual authentication and dynamic key derivation [68]. This method can protect against Evil-twin and Compromised APs, because it is hard to set up a fake authentication server that is protected by strong cryptographic methods. However, the method has to be set up by the administrator. This is difficult to implement, especially in Wi-Fi hotspots; this difficulty allows Evil-twin APs to continue to exist. Another drawback with this method is that the server certificate validation is optional, which may allow the authentication server to be faked by capturing the four-way handshake messages [69], [70].

Web-based Authentication is sometimes used in colleges, cafes, airports, malls, and hotels. In this type of authentication, the user is first directed to a captive portal that asks for credentials or a disclaimer. For instance, many college WLANs use software authentication systems to authenticate students or faculty members on the network. The systems belong to different vendors—either free systems or priority systems—so they are not compatible with one another. In addition, authentication is not related to the network topology, so there is no knowledge of the network's structure. Thus, broadcasts that are sent over WLANs, such as DHCP broadcasts, could be leaked from DHCP requests prior to the authentication of a specific user on the network. This would enable an intruder to break into the network using DHCP requests. The authentication software employed in some colleges uses open WLAN, and the authentication procedure can be done using HTTP. A login webpage is

used to force the user to enter their username and password to authenticate their identity. The authentication process depends on the firewall to redirect the HTTP requests to the login webpage and block all other requests. Once the user has provided the correct credentials, they are authenticated and authorized to access the network resources [71]. The problem with the open nature of WLANs and web-based authentication is that broadcasts such as DHCP frames can be seen by anyone in the network, even if they are not authenticated on the network or authorized to access the network resources. The broadcast frames can be seen by unauthorized users using tools such as Wireshark[8] or tcpdump[9]. This method cannot protect against all RAPs, because it is easy to clone the login webpage and capture users' credentials using tools such as Airsnarf[10]. This method does not provide mutual authentication, whereby the user and the access point authenticate each other; it can authenticate the user, but not vice versa.

VPNs are used to connect to the Internet securely from unsecure environments. To implement a VPN, a tunnel is created over the IP. For example, OpenVPN is open-source software that uses SSL [72]. This method cannot protect against all types of RAP, because the security of VPNs is not satisfactory, especially for portable devices. There are several unsolved attacks that target SSL, such as certificate-based attacks. Thus, it is likely that the VPN session will be aborted because of sinking management packets, forcing the connection to return to the unsecure environment.

[8] A network protocol analyzer.
[9] A command-line packet analyzer.
[10] A utility to set up RAPs.

IEEE 802.11w amendment protects the management and control frames once the session key has been established after the key management exchange. Because the deauthentication and disassociation processes are protected, it is unfeasible to forge the deauthentication/disassociation frames. However, there are some issues regarding the deployment of this standard. Problems with upgrading the firmware and hardware mean that millions of WLAN devices must be changed to become compatible, so most WLANs do not currently implement the 802.11w standard.

3.2 Classification of RAP Detection Approaches

Because the aforementioned countermeasures do not protect against all RAP types, several novel approaches have been proposed by researchers. Some existing approaches use fingerprint techniques to detect the RAP. A device fingerprint aims to stamp a target device using one or more characteristics via its wireless traffic. Fingerprinting can be used for network monitoring, identification, or IDSs. It is triggered either by actively sending traffic to a target device, or passively observing the traffic generated by the target device [73]. Fingerprinting uniquely identifies devices on a WLAN without using identifiers that can be easily spoofed, such as IP addresses and MAC addresses [74]. Some approaches require standard modification, whereas others solve one type of problem. As most techniques focus on detecting Evil-twin APs, we split this section into six categories, two for Evil-twin AP solutions, one for Unauthorized AP solutions, one for deauthentication/disassociation attacks, and one for solutions that detect more than one RAP type. All forged first message approaches require protocol modifications. We do not consider these here, as this survey is

focused on approaches that do not require protocol modifications.

3.2.1 Coexistence Approaches

This subsection introduces approaches that solve the Evil-twin Coexistence sub-type, as classified in Table 3.4. This sub-type seeks to insert an RAP into the WLAN simultaneously with the legitimate AP. In [4], a timing-based scheme was presented that detects RAPs that are injected through a Linux-based machine. In the attacking scenario, the RAP can change its identity by masquerading as the legitimate AP by spoofing the legitimate AP's MAC address and SSID. The RAP then deceives users into connecting to it by increasing its signal strength, and then launches several attacks on the users' machines. The scheme exploits the expected two hops that occur when the user connects to the DNS server.

The authors of [4] used RTT to determine whether or not the given AP is legitimate. The RAP is detected because it relays the traffic to the DNS server via the actual AP. Therefore, the delay results from the two hops that occur between the user and the RAP, instead of the permanent one-hop process. However, the proposed solution needs further investigation, because the authors focused on only one specific cause of the delay in a WLAN. There may be various reasons for such a delay, including (but not limited to) the WLAN's exposure to interference and collisions. Thus, this scheme is neither accurate nor robust, especially in highly traffic-loaded WLANs. Additionally, the proposed technique is more likely to detect the hotspot's AP as an RAP.

Table 3.4: Coexistence techniques

Technique	Source	Year	Accuracy	**P**assive/**A**ctive	No Protocol Modification	Wire**D**/Wire**L**ess/**H**ybrid	**D**edicated/**B**undled	No Special Hardware	Dataset Size
DNS Server two hops	[4], [55]	2009, 2011	60%	A	✓	L	D	✓	2
ETSniffer	[19], [54]	2010, 2012	TPR = 99%, FPR = 1%	A	✓	L	D	✓	NA
WiFiHop	[75]	2011	TPR [a] = 98%, FPR [b] = 0.1%	A	✓	L	D	✓	NA
Authentication + SVM	[76]	2006	86%	A	✓	L	D	✓	5
Duplicate RSSI	[77]	2012	97%	P	✓	L	D	✓	NA
Active Behavioral	[78]	2008	NA	A	✓	L	D	✓	5
Client-side	[79]	2012	NA	P	✓	D	D	✓	NA
Cipher Types	[80]	2012	NA	P	✓	L	D	✓	NA
RAPiD	[81]	2010	NA	P	✓	D	D	✓	NA
Time Interval	[82]	2014	NA	P	✓	L	D	✓	2

[a] True Positive Rate
[b] False Positive Rate

An approach called WiFiHop, in which test packets are actively sent to see if the RAP relays the packets on a different wireless channel, has been proposed [75]. The authors of [76] used SVM to train and validate the precise timing measurements related to the authentication procedure to distinguish fingerprints. This method achieved an accuracy rate of 86%, but the validation considered only five APs. This technique also requires the use of another device to monitor the authentication sequences.

Kim et al. [77] simulated the launch of an RAP while the attacker's device has more than one RSSI. Detection can be achieved using the deviation between the two APs' received signal strength. However, this approach depends on the scenario in which the RAP relays traffic to the actual AP, which is not always the case. Bratus et al. [78] used an active behavioral fingerprinting method adopted from TCP/IP fingerprinting. This approach is implemented by network discovery and security auditing tools like Nmap[11], and applies an active request–response technique. This approach sends a request frame, and then waits for the response in order to determine how the devices react to fragmented or manipulated frames. This technique has the drawback of using active detection, which can be avoided by most attackers. In addition, this technique can interfere with regular WLAN traffic.

Nikbakhsh et al. [79] proposed a multi-step approach to detect RAPs. If two APs broadcast the same SSID and MAC address, the approach checks whether the IP addresses are the same, then compares the trace routes. It is unlikely that the same trace route will be found, because having the same IP addresses at the same time would cause an IP address conflict. Thus, the only possible situation is to have the same IP addresses and different

[11] Free security scanner for network exploration and hacking.

trace routes, which is a result of IP spoofing. This approach cannot deal with such a condition, as it cannot determine which AP is authorized and which is unauthorized.

A second possibility is that there are different IP addresses. The method proposed by Nikbakhsh et al. then calculates the network IDs using different IP classes to compare the IP addresses. If the method finds that the network IDs are identical, the APs are definitely in the same WLAN, which is considered a result of load balancing in the WLAN. In this situation, large organizations use more than one AP to cover the whole WLAN. Thus, the IP addresses of the APs are different, but the network IDs are similar, so the proposed solution marks this situation as safe. Another possibility is that there are different network IDs and different IP addresses. In this case, the approach triggers the trace route for both APs to determine whether there is an extra hop, which would signify that the Evil-twin AP relays packets to the legitimate AP. The last possibility is that network IDs, IP addresses, and routes are different. In this situation, the attacker uses his AP to broadcast the same SSID as the legitimate AP. This situation cannot be handled by this approach, as it cannot determine which AP is legitimate. That is, the approach of Nikbakhsh et al. cannot protect against the Replacement sub-type, as it only detects the Evil-twins that relay packets to a legitimate AP.

Chumchu et al. [83] used the data rates and modulation types to differentiate between legitimate and rogue wireless devices. Important information from PLCP metadata is extracted to detect the rogue devices. The data rates and modulation types rely on a rate adaption algorithm, and are difficult to spoof because they belong to the physical layer. The problem with this approach is that it is limited to the small number of modulation types and

data rates that can be used by the 802.11 standards. There is a high probability that hackers will use similar data rates and modulation types as one or more of the genuine wireless devices in the WLAN.

Chae et al. [80] used the authentication and cipher types of the AP to detect RAPs. Their method stores information on the authorized APs, such as SSID, authentication type, and cipher type, in a database. It then sniffs the beacon frames and compares the parameters with those in the database. If the information does not match that of the authorized APs, an alert is triggered. This approach is designed to be implemented on the client side for protection in airports or malls. However, it is not practical, because all Wi-Fi hotspots in airports and shopping malls are restricted to open authentication (i.e., no other authentication types are used in hotspots) and have only one cipher type.

Szongott et al. [84] combined parameters such as SSID, BSSID, supported authentication, key management, and encryption schemes to detect mobile Evil-twin APs. They also used cell tower information as an environment identifier. Finally, they used the location of the device, as determined by the Google Play services API or through Android's location API. If the user selects a WLAN that is not in the database, no warning message is needed. If the SSID is known, but the BSSID of this AP is not in the database, a warning message is triggered. In this situation, the user has two options. If the user trusts the AP, a profile of this AP is created in the database; otherwise, the connection process is dropped and no information is stored. The other parameters are used to determine the location of the mobile Evil-twin AP. This approach is similar to TOFU, a method used in contexts such as SSH that depend mainly on the user. This method can only detect mobile Evil-twin attacks.

It cannot detect Evil-twin APs that share the Internet with existing legitimate APs, and cannot locate other devices such as laptops or iPhones, because it depends on applications that are related to Android.

Qu et al. [81] proposed an indirect RAP detection approach, known as RAPiD, which uses the Local Round Trip Time (LRTT) of TCP packets to measure the delay. This approach is similar to several other approaches that assume any delay is a sign of RAPs. However, WLANs have two other main reasons for the delay: interference and collision. Kao et al. [82] proposed an approach based on the beacon time interval deviation. The approach takes advantage of the fact that the AP sends a beacon frame approximately every 100 ms, and the time interval between two consecutive beacon frames can be measured to identify suspicious activity. However, it is difficult to predict the time interval between two consecutive beacon frames. Additionally, this approach does not scale in real-life scenarios, because 802.11b, 802.11g, and 802.11n WLAN devices interfere with one another and Bluetooth and microwave ovens cause more interference and collisions in the frequency band. Collecting information from distributed sensors in large organizations would also be a problem, as the time interval would be different from sensor to sensor based on the distance to the AP.

3.2.2 Approaches that handle all Evil-twin sub-types

An overview of the approaches that solve both the coexistence and replacement Evil-twin sub-types is presented in Table 3.5. The authors of [20] combined ISP-based detection and timing-based detection to detect Evil-twin APs. A hotspot's AP must have

a gateway with a global IP address to provide Internet to wireless users. A block of IP addresses is given to the ISP by IANA[12], so the ISP provides a unique global IP address to customers who subscribe to this service. Information in each global IP address, such as the name of the organization, location, and assignment date, is publicly available on various websites. The proposed approach sends a request to one of these servers, and waits for the reply to obtain important information such as the source address of the AP, ISP information, and location. It was found that the hotspot APs that are connected to the same router share the same global IP address or the same ISP. The authors used the information obtained from the public servers to distinguish legitimate APs from Evil-twin APs. ISP-based detection cannot identify Evil-twin APs that share an Internet connection with one of the legitimate APs, as the Evil-twin AP uses the same Internet service, which cannot be differentiated from that of the legitimate AP. Thus, the authors developed another detection method called timing-based detection to detect Evil-twin APs that share the Internet with one legitimate AP. This approach uses active probing, which can add traffic to WLANs.

The work in [85], [86], [87] requires the modification of 802.11 standards or protocols. The authors of [85] introduced a protocol entitled "Secure Open Wireless Access", which adopts the well-known SSL protocol to distribute certificates. The SSID of a given access point is considered a unique string, and is associated with a certificate by a trusted CA. The association between the certificate and the unique string can be used to authenticate the AP operator. The authors of [86], [87] proposed an EAP-based authentication method, referred to as the Simple Wireless Authentication Technique (EAP-SWAT). This utilizes the SSH's trust-on-first-use approach, whereby trust is certified for the first connec-

[12]The authority in charge of managing global IP addresses.

Table 3.5: All Evil-twin techniques

Technique	Source	Year	Accuracy	**P**assive/**A**ctive	No Protocol Modification	Wire**D**/Wire**L**ess/**H**ybrid	**D**edicated/**B**undled	No Special Hardware	Dataset Size
CETAD	[20]	2014	95% [a]	A	✓	D	D	✓	3
SOWA	[85]	2011	NA	A		H	B		NA
EAP SWAT	[86], [87]	2008, 2010	NA	A		H	B		NA
Clock Skew	[88]	2010	90%	P	✓	L	D	✓	41
Clock Skew	[89]	2010	NA	P	✓	L	D	✓	2
Clock Skew	[18]	2014	NA	P	✓	L	D	✓	388
Clock Skew + Temp	[90]	2014	TPR = 90%, FPR = 10%	P	✓	L	D	✓	12
Adjacent Channel	[91]	2015	NA	A	✓	L	D	✓	60
Probe Request Stimuli	[91]	2015	NA	A	✓	L	D	✓	60
Radio Frequency	[92], [93]	2006, 2012	99%	P	✓	L	D		130

[a] For timing-based approach, the average of two results 98% and 92% is calculated to fit into our classification

tion to the AP. Subsequent connections to the AP are ensured to be authenticated by the coexistence of the certificates. For deployment reasons, techniques that require standard or protocol modifications are not ideal solutions. It is impossible to deploy the protocols in [85], [86], [87] because it is difficult to change the drivers and firmware of the supplicants and APs.

Some researchers have focused on hardware fingerprinting to detect RAPs based on the characteristics that uniquely identify the WLAN device. The authors of [88], [89] proposed a clock skewing approach that extracts the TSF timestamp from beacon frames. In addition, the authors compared the beacon frame timestamp generated at the AP with the inter-arrival time of the frame at the user station. This technique is not robust because of variations in the WLAN medium that are susceptible to delay, especially in high-traffic WLANs.

The authors of [18], [90] applied the time skew method using TSF to differentiate between hardware- and the software-based APs. They only detect RAPs that are generated from airbase-ng-based RAP tools, and cannot detect RAPs that are generated by other tools. The authors of [91] used a method called active probing on adjacent channels, which, as the name implies, is an active technique. IEEE 802.11 g/n and some other existing technologies such as Bluetooth operate in the 2.4 GHz band for compatibility purposes. The protocols require channel separation of 16.25–22 MHz, but the problem is that the channel center frequencies can only be separated by 5 MHz, which causes adjacent channels to overlap. It is impossible for WLAN devices to receive a single frame that is not sent on the same operational channel on which this WLAN device operates. It was found that software-based APs

treat these frames in a different way to hardware-based APs. Several probe requests were sent on the operating channel and adjacent channels of 30 hardware-based APs and several software-based APs to examine how probe request frames were treated. It was noticed that hardware-based APs send probe responses on the same operational channel, whereas software-based APs respond to both the operational channel and the adjacent channel.

The authors of [91] proposed another approach called Malformed Probe Request Stimuli. The Address 1 field is set to contain the destination MAC address (i.e., the MAC address or broadcast address of the AP). The Address 3 field is always set to the BSSID; therefore, it is only relevant to IBSSs such as ad hoc or mesh networks. Because the protocol in infrastructure mode states that the BSSID is the AP's MAC address, the AP that receives a probe request should reply to Addresses 1 and 3, which includes the MAC address of the AP. However, the authors noticed that hardware-based APs do not check the Address 3 field of the probe request, unlike numerous software-based APs. This looks reasonable, because APs are designed to be in infrastructure mode and are not part of an IBSS or mesh network. These two approaches have similar drawbacks to other active probing techniques, namely the sharing of bandwidth with the WLAN devices, which causes interference and delay.

Wei et al. [92], [93] used ACK-pairs to distinguish whether traffic was being generated from the wired or wireless side. The authors used an algorithm known as iterative Bayesian inference to acquire a maximum likelihood approximation. Although this approach is effective, it cannot be deployed in real time, because it takes time to converge.

3.2.3 Unauthorized AP Countermeasures

A number of approaches focus on protecting against APs that have been inserted by insiders, as shown in Table 3.6. The authors of [94] proposed an active approach to the detection of unauthorized APs. Their approach has a verifier that is placed on the wired side of the network. This verifier sends test packets to the wireless side of the network. The APs that relay those test packets are detected as RAPs because they are on the wired side of the network and allow the relay of packets to the wireless side. Once an RAP has been detected, its IP address is returned to allow the network administrator to locate the RAP. The verifier was used to monitor the wired side of the network to avoid NAT private IP address problems. The verifier can monitor the active users on the wired side and send test packets to them. If a user who receives this packet is an AP, the packet is forwarded to the wireless side. If the AP uses the WPA or WEP mechanisms, the sniffer on the wireless side cannot reveal the payload of the sent packets. Thus, the authors used the sequence of predefined packet sizes, and employed an active technique to send test packets, although this added an overhead to the shared network medium.

The Shadow Honeypot approach [12] consists of three components: a filtering engine, anomaly detection sensors, and shadow honeypot code. The filtering engine is the first line of protection, responsible for purifying unauthorized wireless traces based on an authenticated list. The authenticated list contains the authorized AP MAC addresses. Any traffic sent from source MAC addresses other than the authorized ones is assumed to originate from an RAP. Traffic from authenticated users is bypassed by the detection engine.

Table 3.6: Unauthorized AP techniques

Technique	Source	Year	**P**assive/**A**ctive	No Protocol Modification	**W**ire**D**/Wire**L**ess/**H**ybrid	**D**edicated/**B**undled	No Special Hardware
Unauthorized Approach	[94]	2009	A	✓	D	D	✓
Shadow Honeypot	[12]	2015	P	✓	L	D	✓
Inter-packet Spacing	[95]	2004	P	✓	D	B	✓
RIPPS	[96]	2008	P	✓	D	B	✓
RTT Approach	[97]	2007	P	✓	D	B	✓
Agent-based	[98]	2003	P	✓	L	D	✓

The traffic that goes through the detection engine is passed to the anomaly detection sensors, which examine the characteristics of the packets and pass legitimate packets to the shadow honeypot stage. The shadow honeypot stage uses popular signatures of worms and attacks and compares them with the network trace. This approach is not very accurate, and is not automated. The authors used different tools to analyze network traffic, an inefficient and time-consuming process. For instance, in the anomaly detection sensor stage, tools such as Wireshark and Ettercap[13] are needed to analyze the network trace and detect RAPs. Additionally, RAPs that have spoofed the MAC address of a legitimate AP have a high probability of passing the other two stages, especially if they send frames that cause a DoS attack. These frames have similar characteristics, and can bypass all of the anomaly detector sensors.

Beyah et al. [95] used the inter-packet spacing to determine whether traffic had been generated from a wired or wireless link. This approach is passive, so it does not add traffic to the WLAN, and can distinguish between wired and wireless traffic. It does not require protocol modification. This approach has a vital drawback, as inter-packet spacing can also be a load on a switch, which might cause this approach to be inaccurate. As the number of switches increase, the accuracy may become an issue. The authors of [96], [97] proposed using the RTT to distinguish between wired and wireless links. The RTT is the time that the TCP/IP session packet pair takes to travel from the router to the host.

An agent based approach has been proposed [98] whereby an agent equipped with a wireless card sniffs wireless frames and returns a packet to the analyzing engine containing

[13] A comprehensive suite for MITM attacks.

information about new APs. The analyzing engine has an authorized list of legitimate APs, so the information corresponding to new APs is checked against the authorized APs to determine suspicious nodes. This type of approach depends completely on the MAC addresses of the APs, which can easily be spoofed.

3.2.4 De-auth/Disassociation Countermeasures

The security standard of 802.11 series WLAN is IEEE 802.11i. This was ratified in 2004, and provides data confidentiality, integrity, and mutual authentication in the MAC layer. It uses 802.1x for authentication and access control, and a four-way handshake for key management and distribution. However, there are some weaknesses in WLANs related to the fact that the management and control frames are unprotected. DoS attacks in WLANs can mainly be classified as deauthentication/disassociation attacks [99], [100] or four-way handshake memory/CPU DoS attacks [101].

The deauthentication and disassociation frames are management frames [102]. They can easily be forged by an adversary if IEEE 802.11w is not implemented, because management frames are not protected. An adversary can spoof the MAC address of a legitimate user, either a supplicant or an authenticator, and send either deauthentication or disassociation packets on behalf of that user to disassociate or deauthenticate the victim. More harmful attacks can be launched by broadcasting these frames on behalf of the authenticator to all the supplicants in the WLAN by setting the destination MAC address to the broadcast address [103], [104]. Thus, one deauthentication/disassociation frame disconnects all of the supplicants on the WLAN.

The deauthentication and disassociation frames are management frames [102]. They can easily be forged by an adversary if IEEE 802.11w is not implemented, because management frames are not protected. An adversary can spoof the MAC address of a legitimate user, either a supplicant or an authenticator, and send either deauthentication or disassociation packets on behalf of that user to disassociate or deauthenticate the victim. More harmful attacks can be launched by broadcasting these frames on behalf of the authenticator to all the supplicants in the WLAN by setting the destination MAC address to the broadcast address [103], [104]. Thus, one deauthentication/disassociation frame disconnects all of the supplicants on the WLAN.

Table 3.7 lists several approaches to detect deauthentication and disassociation attacks launched by wireless users or the AP. Bellardo et al. [105] applied authentication to all of the management frames by modifying the authentication framework. This might help prevent the deauthentication attacks, but it necessitates an upgrade to the AP and WLAN users' firmware. Authenticating each management frame acquires supplementary cost for the AP and the users, consuming the power resources of portable devices. The authors also proposed a delay to the deauthentication effect. If a deauthentication frame followed by a data frame is received from a victim, the deauthentication frame is discarded. However, delaying the management frames generates problems related to roaming.

Sequence number-based approaches [43], [108], [44], [106], [107] have been proposed by several researchers exploiting the fact that every data and management frame has a sequence number in the MAC header. The sequence number typically is incremented by one when the sending device sends a management or data frame. The sensor captures the

Table 3.7: Deauthentication and disassociation techniques

Technique	Source	Year	Accuracy	**P**assive/**A**ctive	No Protocol Modification	Wire**D**/Wire**L**ess/**H**ybrid	**D**edicated/**B**undled	No Special Hardware
Sequence number	[43], [106], [107]	2004, 2005, 2006	FNR [a] = 0.029%-0.036% [b]	P	✓	L	D	✓
Sequence number	[108]	2003	NA	P	✓	L	D	✓
ANFIS	[44]	2010	FAR [c] = 0.00015	P	✓	L	D	✓
Signalprints	[21]	2006	95.6%	P	✓	L	B	✓
SSFA	[109]	2006	NA	P	✓	L	D	✓
K-means	[16], [17]	2007, 2010	FPR = 0.0351 to 0.0957	P	✓	L	D	✓
GMM	[36]	2008	TPR = 98%, FPR = 1%	P	✓	L	D	✓
throughput + Flood	[41]	2013	93%-99% [d]	P	✓	L	D	✓
Machine Learning	[110]	2014	68%-99% [e]	P	✓	L	D	✓
lightweight Solution	[40]	2008	NA	A		H	B	

[a] False Negative Rate
[b] Based on the location of the monitor node
[c] False Alert Rate
[d] Based on the threshold value, as the threshold increases the accuracy increases
[e] Based on the used classifier

frames from the same MAC address, and if it finds there is a gap between two consecutive frames, it assumes that MAC address spoofing has occurred. These approaches cannot work well when the legitimate station is not sending any frames. In addition, it cannot detect an attacker when it only sends control frames, as control frames do not have sequence numbers.

RSSI approaches [21], [109], [16], [17], [36], [45], [46], [111] can be used to differentiate WLAN devices based on their location. The RSSI is the signal power of the frame, measured at the receiving wireless device. A number of factors play an integral role in measuring the RSSI, such as the transmission power, multi-path and absorption effects, and the distance between the two communicating parties. A wireless device does not ordinarily increase or decrease its transmission power, and so obvious changes in RSSI from the same MAC address are an indicator of MAC address spoofing. Because the distance between the adversary and the legitimate wireless device is significant, an adversary is more likely to be detected. One problem with these approaches is that a smart adversary will increase the transmission power to mimic the legitimate wireless device. Another problem is that it is hard to detect the attack, especially if the adversary is in close proximity to the legitimate wireless device.

Chen et al. [16], [17] proposed an approach based on the K-means clustering algorithm to detect MAC address spoofing in WLANs and wireless sensor networks. The authors assume that the RSS samples form a Gaussian. They assume that the RSS samples at a given period at N-sensors form an N-dimensional vector and the number of clusters is two (i.e., $k = 2$). They then use the Euclidean distance algorithm to compute the distance

between the two centroids and eventually detect any MAC address spoofing. In practice, their approach might not work very well, especially when the hacker and legitimate device are close to each other. The centroids of both devices are close to each other, which makes it hard to differentiate the RSS samples that come from the hacker. In addition, their approach struggles with non-Gaussian data distributions. Finally, one device can form two independent clusters, as we explain in the next sections.

Sheng et al. [36] proposed to profile legitimate device RSS samples using the GMM clustering algorithm. They assume that the RSS samples from a given sender-sensor pair follow a Gaussian and apply a GMM clustering algorithm to detect spoofing. The solution that they propose has some limitations: a non-Gaussian distribution of the RSS samples could occur in real wireless networks because of interference, multi-path fading, and absorption effects. As a result, their approach would not perform well, especially in high traffic wireless networks.

Yang et al. [45], [46] proposed to use the Partitioning Around Medoids approach, also known as the K-medoids clustering algorithm, to detect MAC address spoofing. This algorithm is better than K-means because it is robust against any noise and outliers that the data might contain. However, they have similar assumptions to those in [16], [17]. They assume that there are two clusters (i.e., $K = 2$). They also assume that, under normal conditions, the distance between the two medoids should be small because there is only one cluster at a specific location that is the legitimate device. In contrast, under abnormal behavior, the distance between the two medoids should be large and this suggests the existence of an attacker [45], [46]. This approach has a problem that is similar to the

K-means-based approach, which is that it is difficult to determine the attacker if he/she is in close proximity to the legitimate device because the two medoids are close to each other and the RSS samples are mixed together. In addition, one device can have two independent clusters that could degrade the accuracy of their proposed solution.

The authors of [41], [110] assumed that deauthentication causes some degradation in throughput. Thus, they count the number of frames sent by a certain wireless client, and set a threshold value to detect an attack. Although this assumption might be true, it has some drawbacks. First, it is impossible to detect a single deauthentication attack. An attacker can do many disruptive things with only one frame, such as discovering hidden SSIDs or cracking WEP/WPA-PSK methods. Second, a legitimate wireless station may be marked as an attacker simply because it sends two or more frames, as some devices are designed to send more than one frame to leave a WLAN. Nguyen et al. [40] suggested that the AP and WLAN users employ a secret key to authenticate the deauthentication frames. However, this technique would require the firmware of the drivers and devices to be modified.

Tao et al. [42] proposed a layered architecture named Wireless Security Guard (WISE GUARD) to detect MAC address spoofing using three stages. The first stage is OS fingerprinting, which can be applied to the network layer in the protocol stack. The authors extended the SYN-based OS fingerprinting because it is capable of differentiating the attacker from the legitimate device only if the attacker injects data frames into the network. They utilized the capability information, traffic indication map, and tag information (which includes the vendor information) to extend it. The second stage employs the data

link layer, the sequence number field in particular. They utilized the idea that there could be a sequence number gap between the legitimate device and the attacker consecutive frames. The third stage brings to play the RSS, which belongs to the physical layer; unfortunately, the authors did not explain this stage in much detail.

The authors established some rules to detect the MAC address spoofing. They used a simple and yet effective technique to combine the outputs from the three stages. Every stage outputs either normal or abnormal states of every upcoming frame. They then combined the outputs to evaluate how severe the suspicious frame is; if the analyzer finds the outputs of more than one stage to be abnormal, the alert is triggered. If the OS fingerprinting stage alone is abnormal, the alert is triggered. This indicates that the MAC address of the AP is masqueraded, because the OS fingerprinting that the authors used, depends on fields that are vital to the APs such as capability information. Some drawbacks exist in such approaches: most of the spoofing attacks involve control and management frames, and these frames cannot reveal OS characteristics; therefore most of the intrusions in WLANs go undetected. OS fingerprinting also assumes that most of the tools that attackers use are based on Linux based operating systems. This is somehow a valid assumption, but Windows Operating System also provides a capability to change the MAC address of any wireless card in the WLAN. The sequence number techniques have several drawbacks as explained previously, so combining both SN and OS fingerprinting could miss some intrusions.

3.2.5 Countermeasures that Solve Multiple Attacks

The approaches listed in Table 3.8 can protect against multiple RAP types. In [6], [58], a hybrid approach was proposed that works on the wired and wireless sides of the network. This approach includes several centralized and distributed tasks. A frame collector is used to capture frames and filter anomalies, allowing Evil-twin, Unauthorized, and Compromised RAPs to be detected. This approach has two main drawbacks: it uses active probing, and must be bundled with the router or the switch. It is difficult for the router or the switch to divide its work between serving the wireless users by carrying traffic and acting as an IDS.

Companies such as Air-Magnet [115] use wireless sniffing solutions. Sensors are deployed across the whole diameter of the network to gather physical and data link layer information, enabling RAPs to be detected in a distributed agent–server architecture [115], [116]. The collected information contains RF measurements, MAC addresses, signal strengths, and AP control frames. This approach is very expensive, because the analyzer system provided by Air-Magnet costs $3,000 [13], [115].

Vanjale et al. [112] proposed using the SSID, MAC address, and RSSI to detect RAPs. The authors created a profile containing these three parameters for each legitimate AP. This technique first checks the AP SSIDs. If it finds any duplication, then it considers the MAC addresses of the duplicate APs. If both are the same, this is considered a legitimate AP. If different MAC addresses are found, the RSSI is checked. If the difference in RSSIs is less than 10 dB, then the technique considers this AP legitimate. This

Table 3.8: Techniques that protect against multiple RAP types

Technique	Source	Year	**P**assive/**A**ctive	No Protocol Modification	Wire**D**/Wire**L**ess/**H**ybrid	**D**edicated/**B**undled	No Special Hardware	Detect **E**vil-twin, **U**nauthorized, **C**ompromised
RAP	[6], [58]	2007, 2008	A	✓	H	B	✓	E, U, and C
Elimination	[112]	2014	P	✓	L	D	✓	E and U
Multi-Agent	[113]	2010	P	✓	L	D	✓	E and U
DWSA	[114]	2004	A	✓	L	D	✓	E and U

approach is passive and does not require protocols or standard modifications, but it has some drawbacks. The first is that, in reality, it cannot detect Evil-twin APs, because these RAPs can mimic the same SSID and MAC address as one of the legitimate APs. This approach assumes that APs with the same SSID and MAC address are genuine; however, this assumption is misleading. A second drawback is that this approach detects a hotspot's APs as RAPs, as they have the same SSID but different MAC addresses.

Sriram et al. [113] proposed a multi-agent solution that can detect Evil-twin and Unauthorized RAPs. This approach has two important components, namely a master agent and a slave agent. The master agent is used to regulate the authorization processes of the WLAN, while the slave agent is used by the master agent to identify active APs in the WLAN. The slave agent is connected to an AP to obtain important information such as SSID, vendor name, MAC address, and channel number. This information is sent to the master agent and compared with information on an authorized list. However, this approach depends on parameters that can be easily spoofed by many Evil-twin tools. Such approaches use an agent equipped with a wireless card to sniff wireless frames and return a packet containing information about new APs to the master agent. The master agent has an authorized list of legitimate APs, and checks the new AP against the authorized APs to determine suspicious nodes. This type of approach is heavily dependent on the AP MAC addresses, which are easy to spoof.

In [114], a Distributed Wireless Security Auditor (DWSA) was proposed. This approach uses both Linux and Windows-based implementations to provide network administrators with continuous wireless assessments. It also uses trusted wireless clients as

distributed sensors to find anomalies throughout the WLAN. DWSA provides periodic security reports, and detects and locates RAPs using 3D trilateration. This approach can detect Evil-twins and Unauthorized RAPs.

Companies such as NetStumbler [117] use wireless packet analyzers on laptops or hand-held devices to detect RAPs. That is, IT personnel physically walk through the halls of an organization or university to search for RAPs. This technique is time-consuming and ineffective, because the scan is performed manually. Additionally, IT employees should upgrade the detection devices to be able to work on different frequencies. Furthermore, the scan can be evaded if the hacker simply unplugs the RAP as the detection is taking place.

Various techniques [118], [119], [120], [121] use a scan from a central location to achieve enterprise-wide coverage. Several dedicated sensors are distributed with the help of one or more legitimate APs to scan beacon frames from surrounding areas. Information on the surrounding APs is sent to a central unit for further analysis under the prevailing security policy. The problem with these techniques is that each sensor only scans one frequency, and some sensors only cover one channel. Another problem with some techniques is that they detect neighboring APs as RAPs.

The authors of [3] used several light machine-learning algorithms that could classify the four classes that they studied for WLAN attacks. The best performing classifier was J48, with an accuracy of 96.19%, when using all the 156 feature set. This algorithm takes, about 3921.68 seconds. The authors then reduced the dimensionality of the data-set and picked the best 20 features to improve accuracy and reduce time. They were able to increase the accuracy of the best performing algorithm to 96.2574% and decrease the time of that

algorithm by 568.92 seconds.

In this research we will not survey the wired IDSs researches [27], [122], [123], [124], [125] as they are limited to detect network, transport, and application layers attacks. We will only consider the wireless IDSs that are closely related to our research which work on the two layers that are only available in WLANs which are physical and data link layers.

3.3 Road Map and Future Directions

The simplicity of configuring an RAP creates a real security threat to WLAN devices. There are several existing techniques that can detect RAPs, but they are inefficient and often inaccurate. Some techniques require the active addition of traffic to the WLAN, whereas other techniques require protocol modifications. The current techniques have several drawbacks, as listed in Table 3.9. Early wireless-side solutions detected Evil-twin APs by examining SSID and MAC addresses to differentiate legitimate (authorized) APs and locate the RAPs. The wired-side solutions locate RAPs using switch port mapping, but do not have an integral authorization method as they depend only on switch port policies. Furthermore, it is not possible to detect an RAP that is attached to a legitimate AP. The wired-side solutions must require authorization techniques other than the switch port policies.

Table 3.9: Strengths and weaknesses of existing techniques

Technique Type	Strengths	Weaknesses
Unautomated Wireless Solutions	Passive Minimal infrastructure is needed	Can be evaded easily Requires considerable effort and time Sensors must perform on every channel
Wired-Active Probing	Does not depend on wireless frequency	Active RAP might not respond to packets Only depends on switch port policies
Hybrid	Passive Can detect most RAP types	Can be evaded from the wired side
Timing approaches	Passive Does not depend on wireless frequency	Necessitates samples on wired and wireless Assumes wired link faster than wireless Could be evaded from insiders
Identification approaches	Passive Does not depend on wireless frequency No samples from wired and wireless Link speed is not important	Could be evaded from insiders

The road map in Figure 3.5 shows how the detection of RAPs has evolved from manual scanning by walking through halls to automated WIDS. Based on our survey, it is clear that future solutions should have numerous characteristics. A complete solution to the RAP problem should be able to detect all RAP types. A passive approach is preferable, as this will not increase the traffic on the WLAN. In addition, approaches that require protocol modifications or additional special hardware, besides sensors, should be avoided, because deploying modifications can be difficult, supplying new hardware is costly, and implementation may cause incompatibilities. An approach that is implemented on the AP is disadvantageous, as it requires the detection task to be shared with the serving of wireless traffic. An ideal approach would allow complete coverage of a WLAN, including all possible channels and frequency bands. For robustness, a suitable approach should not rely on higher-layer protocols such as TCP ACKs, because this will delay detection and is ineffective against deauthentication/disassociation and forged first message attacks, which depend on management frames rather than higher-layer protocols. Finally, a well-built approach

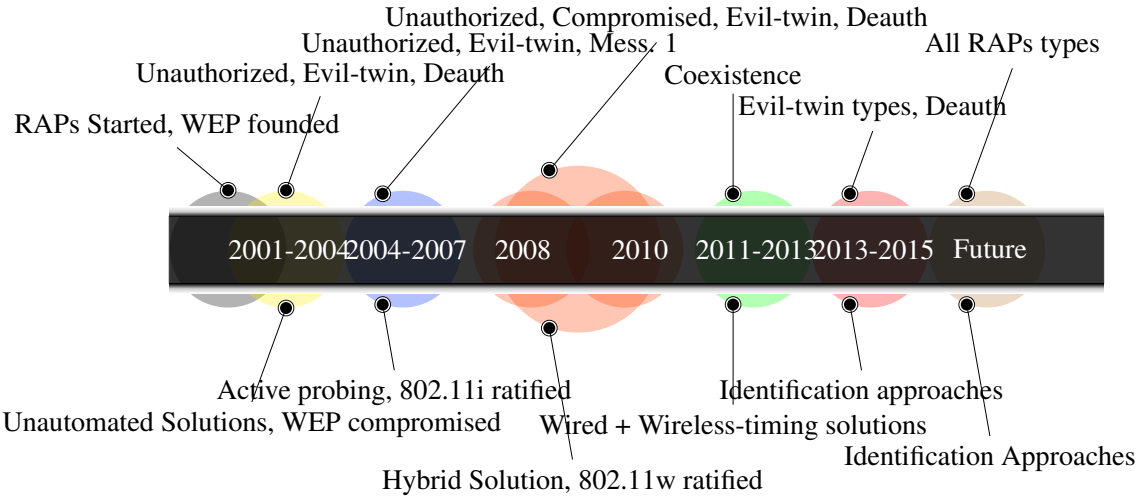


Figure 3.5: Timeline of existing techniques

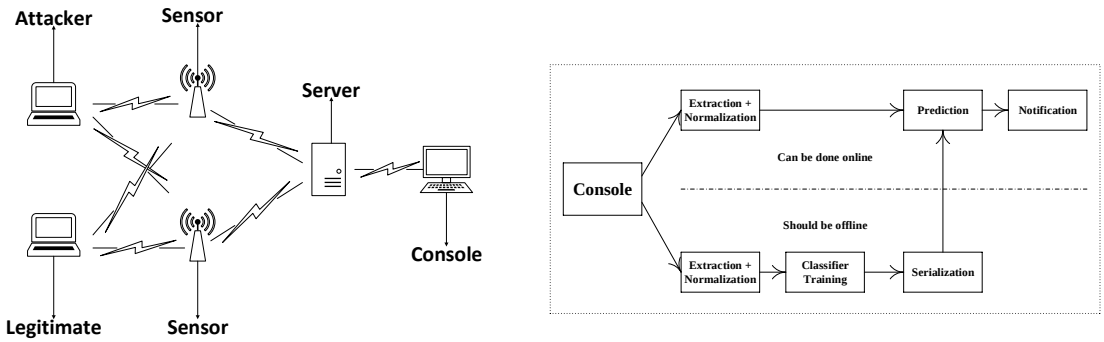
should not depend on easily spoofed identifiers such as MAC addresses or IP addresses.

CHAPTER 4: RESEARCH PLAN

RSS has been adopted by researchers for localization for several years because of its correlation to the location of a wireless device [126], [127], [128], [129], [130], [131]. The goal of localization is to focus on RSS samples of a single device. In contrast, in spoofing detection, it is sometimes difficult to distinguish between two devices at different locations that claim to be the owner of a specific wireless device through spatial information alone, especially when they are in close proximity. We exploit the fact that RSS samples at a specific location are similar while the RSS samples at two different locations are distinctive. To distinguish an attacker, we should first develop the characteristics of normal behavior by building a profile of the legitimate device.

4.1 Network Architecture

The network architecture is assumed to be similar to the one that is in Figure 4.1(a) which consists of sensors monitoring the network. Every sensor captures frames from nearby wireless devices. Each sensor sends the important information of the captured packets, as shown in Figure 4.1(a), to the server for global detection. The console receives the



(a) Network Architecture (b) Profiling and Detection

Figure 4.6: Network Architecture and Profiling

packets, normalizes the RSS samples using the timestamps or sequence number, combines the packets, and constructs the sample. Each sample contains the information of the same packet from both sensors.

4.2 Profiling based on Random Forests

The proposed framework involves two stages: the offline stage and the online stage. In the offline stage, the legitimate device profile is built. During profiling, we label the legitimate device RSS samples for the training set as 0 and all possible other locations as 1 to construct a profile of the legitimate device. We train the classifier on 50% of the data for each combination (this can be done once per new environment or periodically). We test on 50% of the unseen data to evaluate our predictor. Once we are satisfied with our predictor, we can serialize it, as shown in Figure 4.1(b), to predict new unseen data. After serialization, the training procedure depicted in the lower part of the figure is not necessary for real-time prediction. Thus, in the online stage, any new packet can be fed immediately

to the predictor. The predictor then predicts if the packet comes from a legitimate device or not. If it finds that the packet is coming from a suspicious device, an alert is triggered.

Let x denote the RSS sample and C denote the class, so that

$$C = \begin{cases} 0 & \text{if } x \text{ is genuine} \\ 1 & \text{if } x \text{ is suspicious} \end{cases}$$

Data points are denoted by a vector

$$v = (z1, z2, ..., zm) \tag{4.1}$$

where z is an integer representing the signal strength of each frame in the signal space.

dataset d can be represented as

$$d = (x_1, y_1), ..., (x_n, y_n) \tag{4.2}$$

where d = *20000* for each combination in (4.2), x_i is the RSS sample and y_i is its label.

and $x_i \in N - dimensional$

where N-dimensional is feature vectors having RSS samples captured by each sen-

sor (e.g., the first feature is the RSS samples captured by the first sensor, the second feature is the RSS samples captured by the second sensor and so on).

We used the Python library [132] in our experiment to train and test our detection method. Algorithm 1 shows the training set using the Random Forests ensemble method. Random Forests uses a specified number of trees (e.g., 100) to perform the whole procedure. Each tree works on a different subset of the dataset randomly to create the ensemble [133].

Algorithm 1 Training using the Random Forests algorithm

1: **for** t = 1 to F **do** $\triangleright$ $F = 100$

2: Uniformly render a bootstrap sample $\mathbf{Z}^*$ from d

3: Random Forests tree T_t increases bootstrapped data $\mathbf{Z}^*$ in size by performing the following steps:

- At each node choose r features randomly
- Choose the best possible feature $\triangleright$ $x_i \in N - dimensional$ as stated previously
- Split into two child nodes using the best split-point $\triangleright$ $r \leqslant N$

4: **Output:** Trees ensemble $\{T_t\}_1^F$

To detect MAC address spoofing, we used the prediction ability of Random Forests after serialization to predict unseen new samples, as indicated in Algorithm 2. The new sample is classified as normal or abnormal, if the predictor finds it to be different from the profiled samples.

Algorithm 2 Detection algorithm

1: **for** profiled MAC address frames **do**

2: Predict every sample using the following equation ▷ To predict new data point $\hat{x}$

$$c_R^F = M_v c_t(\hat{x})_1^F \tag{4.3}$$

▷ c_t is the prediction class of the Random Forests ▷ M_v is the majority vote

3: **If** the sample is different from the legitimate device samples

4: **Output:** A rogue device has been detected

4.3 Anomaly Detection

The anomaly detection capability utilizes the one-class SVM to detect anomalies. The capability uses the Radial Basis Function (RBF), because the data shape is non-linear. Also, the capability is total unsupervised (i.e., the samples are not labeled). In the offline stage, the anomaly detection capability uses one-class SVM-RBF to build up the virtual profile of the RSS samples as shown in Figure 4.7, then it passes the samples to the normal profile builder for building the patterns. In the online stage, the network traces are passed and the RSS samples are extracted. Consequently, the anomalies detection takes place to find out if the RSS samples are similar or different from the constructed profile. If it finds it different from the profiles samples, the alert is triggered.

4.4 Applying the Misuse Detection on Public Dataset

The proposed framework (shown in Figure 4.8) uses several machine learning algorithms to build the patterns of both the normal behavior and the intrusions. The patterns

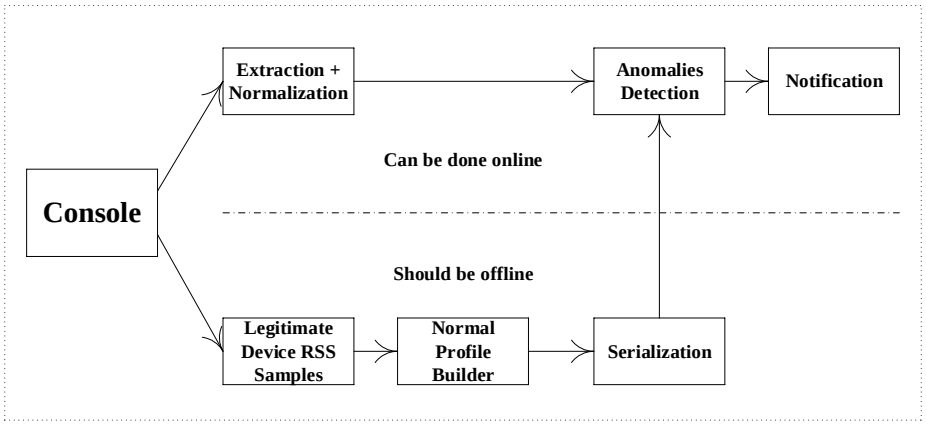


Figure 4.7: Anomaly Detection

of the intrusions are built in the offline stage. The intrusions are classified based on their types in the online stage. Prior to training the framework applies a feature selection capability to choose important features and discard unwanted features. The training includes some algorithms that are fed into majority voting for robustness and to improve the performance. These algorithms are Extra Trees with 20 trees, Random Forests with 20 trees, and Bagging with 10 Decision Trees. After majority voting, the patterns are built by the matching builder for normal samples and intrusions. Once the builder creates the patterns, the patterns can be serialized and fed into the detection capability. In the online stage, the network traces are pre-processed using the features that have been selected by the feature selection capability. After pre-processing, the frames are fed into the detection utility for online detection. The detection utility decides whether the frame is suspicious or not; if it

finds a suspicious one, the alert is triggered.

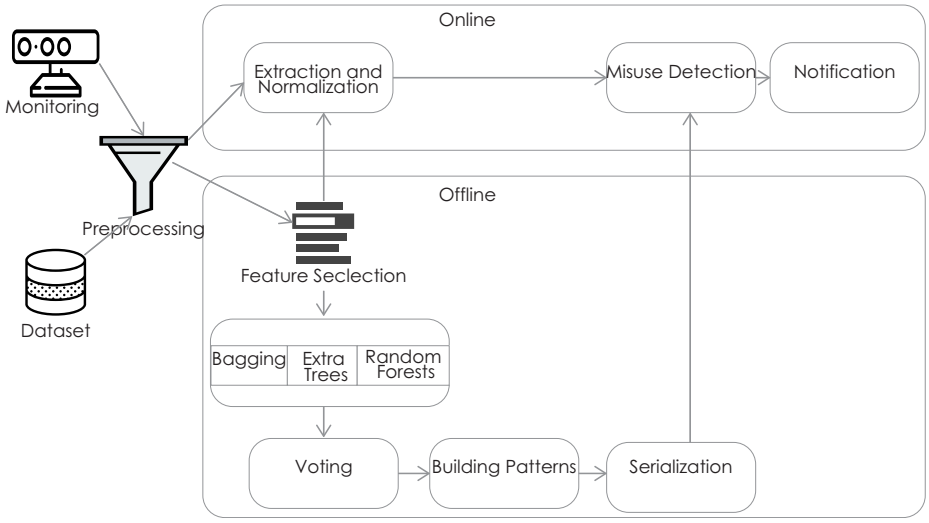


Figure 4.8: Misuse detection

4.4.1 Bagging

Tree Bagging algorithm was found by Leo Breiman in 1996 [134]. Bagging ensemble method consists of predetermined and parallelized classification trees. These trees are grown from bootstrap replications. The randomization of the cut-points is accomplished implicitly through the bootstrap re-sampling.

4.4.2 Random Forests

Random Forests classifier was also introduced by Breiman in 2001 [30]. Random Forests ensemble method is constructed using collections of weakly-correlated decision trees. A bootstrap sample of the training set is used to train each tree in the forests. The best split is chosen at each node from a random subset of the features. This procedure

guarantees that each tree uses independent features from the training samples. Thus, it helps reduce the statistical correlations on the rest of the trees.

4.4.3 Extra Trees

Extra Trees was found by Geurts et al. in 2006 [135]. The ensemble method utilizes the top-down procedure to construct an ensemble of unpruned decision trees. The cut-points selection is carried out fully at random to provide the best split of the nodes. Extra Trees algorithm grows the trees by utilizing the entire learning sample instead of a bootstrap replication.

4.4.4 Majority Voting

Majority Voting is one of the most popular voting methods along with Plurality Voting and Weighted Voting [136]. Majority Voting has been used by several researchers, utilizing the base classifiers to obtain better results. There are some advantages in combining several classifiers (such as increasing robustness, obtaining better accuracy, and heavily built generalization) [137], [138], [139], [140], [141]. The vote for one class is carried out by each base classifier, and the final class label is the one that receives more than half of the votes. If there is no class label that receives more than the half of the votes, the majority vote technique makes no prediction (i.e., a rejection option is given) or one of the base classifiers option is explicitly selected. In this article, we first used the best performing classifiers to get strong generalization. Then, we used majority voting technique to get

better accuracy.

Suppose a set of N classifiers (i.e., ensemble methods in our case) are given $\{e_1, ..., e_N\}$ and our aim is to incorporate e_i's to predict the target of a given sample from a set of t targets $\{c_1, ..., c_t\}$.

Suppose that for a given sample s, the outputs of the ensemble e_i can be given as $t-dimensional$ target vector $(e_i^1(s), ..., e_i^t(s))^\top$

Where $e_i^j(s)$ is the output of e_i for the class target c_j.

The $e_i^j(s) \in \{0, 1\}$, which is one if e_i predicts c_j as the class target and zero otherwise.

The popular majority voting technique for binary classification problems (where every ensemble method votes for a target) is introduced in this subsection. In this technique, the target that gets more than half of the votes would be selected. However, if none of the targets gets more than half of the votes, either a rejection option is given (which indicates no prediction is taken) or we trust one of the classifiers to predict. So, the target of our method can be assigned as:

$$E(s) = \begin{cases} c_j & \text{if } \sum_{i=1}^{N} e_i^j(s) > \frac{1}{2} \sum_{m=1}^{t} \sum_{i=1}^{N} e_i^m(s), \\ rejection & \text{otherwise.} \end{cases} \tag{4.4}$$

where E is a set of learners (i.e., ensemble methods).

For the binary classification problem, the ensemble decision will occur if at least $\lfloor \frac{N}{2+1} \rfloor$ of the classifiers select the right class target.

$$P_{MV} = \sum_{m=\frac{N}{2+1}}^{N} \binom{N}{m} p^m (1-p)^{N-m} \tag{4.5}$$

where p is the probability to classify the correct class target.

The previous two equations can work if the classification problem is binary. In case the classification problem is multi-class, the plurality voting should be utilized to choose the class target that gets the largest number of votes as the correct target. So, the selected class target should be:

$$E(s) = c_{\underset{j}{\text{argmax}} \sum_{i=1}^{N} e_i^j(s)}, \tag{4.6}$$

It is noticeable that plurality voting does not contain rejection propriety because it should always realize the class target that gets the largest number of votes.

4.4.5 Feature Selection

Some of the frames fields are not necessarily for distinguishing between the legitimate devices' traces and the attackers' traces. Extracting unwanted features adds time overhead and might not improve the performance. Feature selection is a valuable initiative to build IDSs, especially machine learning-based IDSs. Although, the number of features is

definite since it depends on the frame header, many other features can be added artificially to the frames metadata when capturing the frames. However, only some frames fields are crucial to detect the intruders. Some machine learning algorithms are hypersensitive to the number of features; choosing the significant features increases the performance of the IDS and decreases the time. Some researchers reported that choosing the suitable features is hard and time consuming. The usual, prune-to-errors way to in choosing the right features is to let the security expert decide which features are important. A better way to do it is to use the data mining approach to discover important patterns of large data sets. It can build intrusions patterns, which can be used for misuse detection techniques based on classification or can build profiles of normal behavior to detect intrusions by anomaly detection techniques.

Some information might obstruct the classification task, especially in classification problems that consist of many different and connected correlations. Incorrect interrelationships exist in features which affect the detection performance. Some features might be needless or redundant. Furthermore, reducing the features could improve the computation time and the performance of the WIDS. It is impossible for human to discover the complex correlations that exist between features. Feature selection is significant for the WIDS to perform real time prediction, so reducing the features is recommended. Thus, reduction could be done using data filtering by system experts' supervision or by data mining techniques. The former might ignore useful data, so it has to be done with caution.

CHAPTER 5: IMPLEMENTATION AND TEST PLAN

We covered an area of 102 m^2 using 15 locations marked by the red dots in Figure 5.9 to evaluate our proposed method. The distance between any two neighbors is about four meters (from 3 to 5 meters). We tried to simulate the attacker to be at every possible place throughout our test-bed. We placed two sensors, indicated by the triangles, to cover as much ground as possible of the network diameter. We also used some active probing techniques to force the device to respond to specific frames in order to speed up the process of profiling. Each sensor captures enough packets for legitimate device profiling. The total number of combinations is 105; we chose one location to be the location of the legitimate device (e.g., location 1) and picked another location for the suspicious device (e.g., location 2) and ran the test for all other locations (e.g., location 3 to location 15) as attacker against the legitimate device (i.e., location 1) as well. We tested all possible combinations.

5.1 Hyperparameter Optimization

To avoid high variance and determine whether the dataset is sufficient to train a Random Forests classifier of 100 trees, we used the learning curve of one of the noisiest

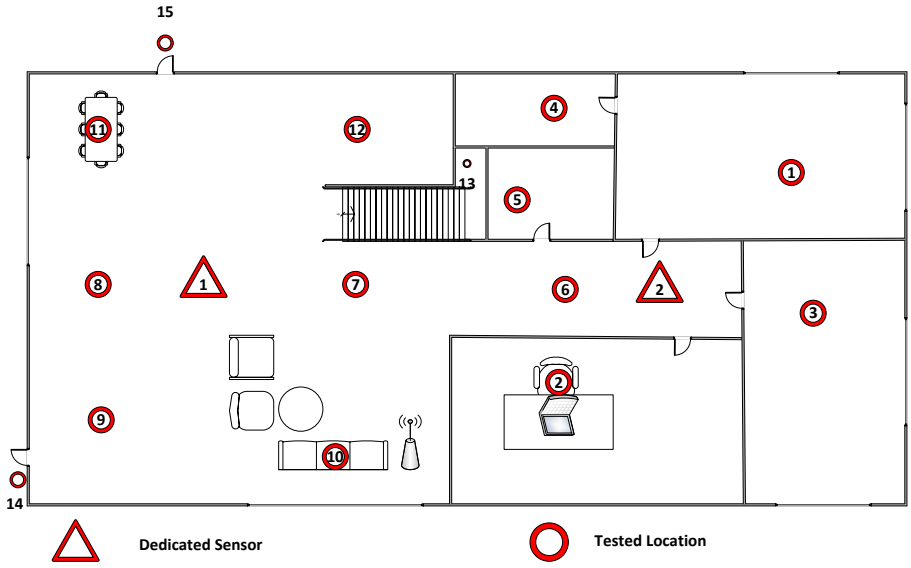
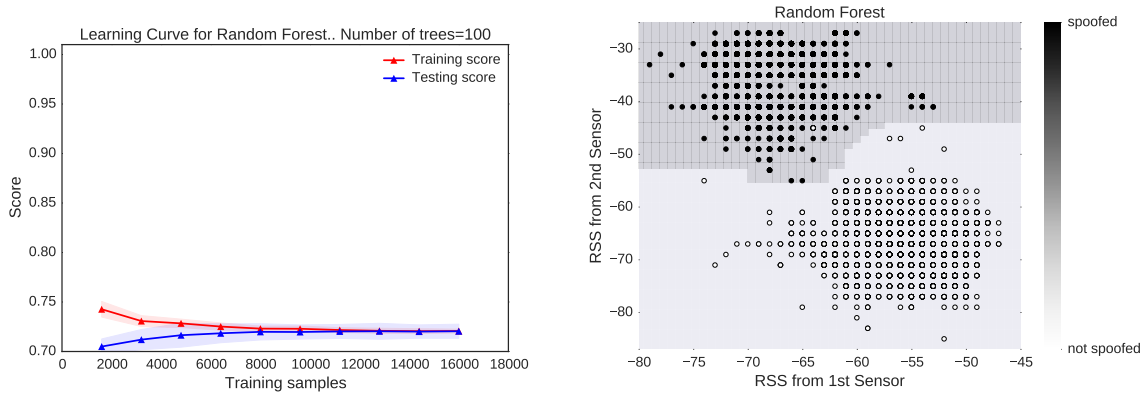


Figure 5.9: Test-bed

datasets, that of locations 6 and 7, where the distance between the two locations is less than 4 m, shown in Figure 5.2(a). We started with about 3,000 samples and determined that we could improve the accuracy and reduce the variance. At about 15,000 samples, the variance was eliminated and stabilized, indicating that a dataset of 20,000 observations is more than enough. Figure 5.2(b) shows how Random Forests of 100 trees separates the data-points when the attacker is 10 m away from a genuine user. The Random Forests ensemble method performed very well in the presence of outliers and can separate data of any shape.



(a) Learning Curve of Random Forests with 100 trees for locations 6 vs 7

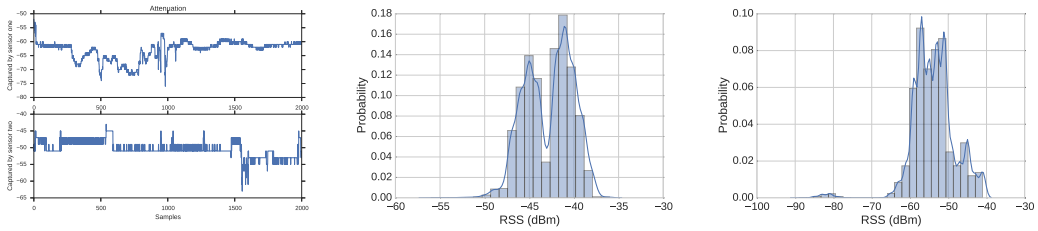
(b) Performance of Random Forests when the attacker and legitimate device are 10 m apart

Figure 5.10: Optimization and data separation

5.2 Signal Strength Attenuation

Figure 5.3(a) illustrates the signal attenuation that signal strength might face in wireless networks. We picked two of the sampled locations to represent this phenomenon and measure 2,000 consecutive packets at each location. One sampled location is close to the first sensor and the other one is close to the other sensor. The two subplots show an attenuation of about 3.4 dB standard deviation (a maximum of -52 dB and minimum of -76 dB) for the first sampled location and a 2.4 dB standard deviation for the second sampled location (a maximum of -43 dB and minimum of -63 dB). It is not rare to see some signal attenuation in our experiments. This phenomenon exists because of several factors such as multi-path fading and obstacles that could make the signal oscillate, especially when there is a significant distance between the sender and receiving device.

The distribution of the data from location 8 at the two sensors is shown in Figure 5.3(b) and 5.3(c). Some researchers state that the distribution of the transmitter and sensor



(a) Signal strength attenuation

(b) Location 8: sensor 1 data distribution

(c) Location 8: sensor 2 data distribution

Figure 5.11: Data distribution and attenuation

pair is Gaussian [16], [17] while other researchers report that the distribution is not Gaussian [47] or that it is not rare to see non-Gaussian distributions of RSS samples [36]. We found that non-Gaussian distributions are not rare and have different distribution shapes and peaks. The distribution of 10,000 RSS samples is shown in the figure. Figure 5.3(b) shows a distribution of data that form two Gaussians with one peak that is slightly greater than the other one (i.e., one device has formed two separate clusters) while Figure 5.3(c) shows a distribution of data with one Gaussian and some sporadic data points that are far away from the Gaussian. This suggests that using clustering algorithms-based approaches [45], [46], [36], [16], [17] can generate many false alerts or cause the Intrusion Detection System to allow large margins that permit attackers to harm the network.

CHAPTER 6: RESULTS AND EVALUATION

To evaluate our proposed solution and compare it with previous work [45], [46], [36], [16], [17], we implemented the four possible GMM kernels because the kernel that [36] used was not indicated in their article. We considered only the best performing kernel (i.e., GMM-Full) for comparison. We first calculated the accuracy of the previous proposed solutions [45], [46], [36], [16], [17] along with our proposed method. The clustering algorithms-based approaches [45], [46], [36], [16], [17] did not work well, as shown in Table 6.1(a), especially when the two locations were close to each other because of the reasons mentioned earlier (see subsection 5.2).

Our proposed method achieved the best accuracy of 94.83. We tested all the detection methods where the distances between the two locations were less than 4 m, as shown in Table 6.1(b), between 4 and 8 m, as shown in Table 6.1(c), and between 8 and 13 m, as shown in Table 6.1(d). When the locations are close to each other, the clustering algorithms-based approaches [45], [46], [36], [16], [17] did not perform well, with a minimum of 47.18% accuracy for Sheng et al. approach [36], as shown in Table 6.1(b). All the techniques did slightly better when the locations were a little further apart, as shown in Table 6.1(c). However, all these methods did very well; our method's performance remains

high when the distance between the two locations increases, as shown in Table 6.1(d).

6.1 Performance Measures

To evaluate our detection method more rigorously, we used the Receiver Operating Characteristic (ROC) curve, shown in Figure 6.1(a), which plots the Detection Rate, that is, the True Positive Rate or sensitivity against the (1 - specificity) or False Positive Rate (FPR). We evaluated our detection method to measure the tradeoff between correct detection and FPR for different distances between the attacker and legitimate device. At 3% FPR, the correct detection rate is about 99% for all combinations in our test-bed. At 12% FPR, the detection rate is 99% when the distance between the attacker and legitimate device is between 4 and 8 m. At 25% FPR, the detection rate is 90% when the distance between the attacker and the legitimate device is less than 4 m, and 100% when the distance is between 8 and 13 m. We also measured the prediction time to see if it is possible to predict the captured frames in real-time. Table 2 shows the average testing time, standard deviation, minimum, and maximum values for 10,000 samples of all the tested locations. The clustering algorithms-based methods, Chen et al. [16], [17], Sheng et al. [36], and Yang et al. [45], [46] are faster than our method. Chen et al. [16], [17] approach is the fastest with times as high as 48 ms. Figure 6.1(b) illustrates the overall performance of our detection method and the existing methods with regard to testing time. Our detection method has a good performance in terms of testing time, with an average of only 155 ms as shown in Table 6.11.

Table 6.10: Detection accuracy by distance between locations

	Chen et al. [16], [17]	Sheng et al. [36]	Yang et al. [45], [46]	Our method
mean	88.9492	87.5902	91.1658	94.8296
std	14.0435	15.2362	11.0422	7.1087
min	53.38	28.61	53.21	71.35
50%	96.08	94.95	96.47	98.81
75%	98.75	99.53	98.76	99.92
max	100	100	100	100

(a) All location combinations (105 combinations)

	Chen et al. [16], [17]	Sheng et al. [36]	Yang et al. [45], [46]	Our method
mean	76.5895	76.3920	80.3875	88.3800
std	15.4416	15.2714	13.5181	8.2278
min	53.41	47.18	53.21	75.88
50%	77.520	70.375	81.345	89.640
75%	89.3675	90.6650	90.9975	94.5825
max	98.56	98.25	98.56	99.77

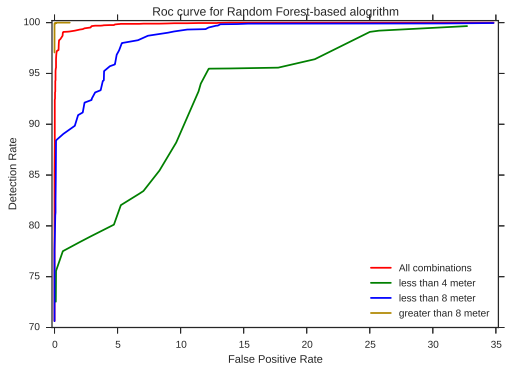
(b) Locations $<$ 4 m apart (20 combinations)

	Chen et al. [16], [17]	Sheng et al. [36]	Yang et al. [45], [46]	Our method
mean	85.7275	82.5584	89.1618	93.1614
std	14.2020	16.0099	10.0814	6.8342
min	53.38	28.61	64.56	71.35
50%	91.360	84.740	92.600	95.610
75%	96.2825	96.5850	96.9475	98.6025
max	99.72	99.91	99.72	99.95

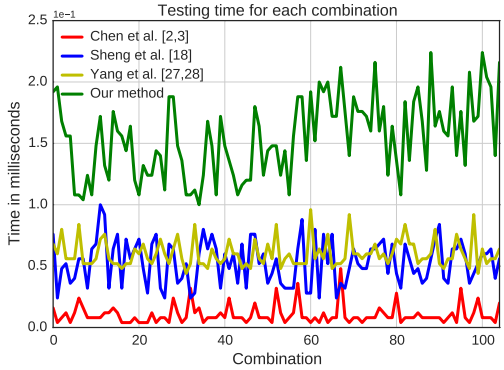
(c) Locations $>$ 4 m and $<$ 8 m apart (44 combinations)

	Chen et al. [16], [17]	Sheng et al. [36]	Yang et al. [45], [46]	Our method
mean	98.4359	98.4527	98.5741	99.7661
std	1.6246	2.3989	1.4843	0.42908
min	94.31	92.04	94.97	98.22
50%	99.09	99.72	99.09	99.95
75%	99.76	99.94	99.76	99.98
max	100	100	100	100

(d) Locations is $>$ 8 m and $<$ 13 m apart (41 combinations)



(a) ROC Curve for the proposed method



(b) Testing time for the tested methods

Figure 6.12: ROC Curve of the proposed method and testing time of all the methods

Table 6.11: Testing time for all location combinations

	Chen et al. [16], [17]	Sheng et al. [36]	Yang et al. [45], [46]	Our method
mean	0.010400	0.053219	0.060190	0.154705
std	0.007718	0.017691	0.010918	0.031848
min	0.004	0.024	0.044	0.100
max	0.048	0.100	0.096	0.224

6.2 Discussion

RSS measurements can be utilized to differentiate wireless devices based on location. Some factors play a vital role in measuring the RSS, such as multi-path fading, absorption effects, transmission power, and the distance between the transmitter and the receiver. Our experiment shows multiple situations where the data forms different shapes and peaks. This is probably because WLAN devices interfere with one another. In addition, microwave ovens and Bluetooth might cause more collision and interference in the frequency band. Thus, our proposed method is very effective because (unlike the previous solutions [16], [17], [36], [45], [46] that could deal with the data if it is only Gaussian distributed) our method could pick the data of any shape. The overall accuracy of our pro-

posed method is 94.83% of all combinations which outperforms the previous solutions: the overall accuracy of Chen et al. [16], [17] solution is 88.95; the accuracy of Sheng et al. [36] solution is 87.59%; and the accuracy of Yang et al. [45], [46] solution is 91.17%.

We tested the proposed method where the distances between the genuine device and the attacker is less than 4 m, from 4 to 8 m, and from 4 to 8 m. The longest distance between any two locations in our test-bed is about 13 m. Although we did not test any two locations where the distance is more than 13 m, we believe that the accuracy would be perfect as the distance between the attacker and the legitimate device increases to more than 13 m. We also did not test different types of antennas such as directional or beam antennas, because this research assumes that the attacker uses an omnidirectional antenna, so more sophisticated attacks might remain undetected.

The sensors placement is significant to determine the difference between the profiled legitimate device samples and the masquerader frames. Figure 6.13 shows how important the features after training are at determining the two locations for three different combinations (note that understanding feature importance is a capability that is provided by almost all of the ensemble methods). The first feature comprises the RSS samples captured by the first sensor, and the second feature consists of the RSS samples captured by the second sensor. The figure shows which sensor determines most of the samples of locations 1 and 14. It appears that the two sensors are close: about 51% are determined by the first sensor and 49% by the second sensor. In this case, the distance between the attacker and the legitimate device is about 12 m. The legitimate device (i.e., location 14) is 3 m from the first sensor. The hacker (i.e., location 1) is about 9 m away from the first sensor and

about 3 m away from the second sensor.

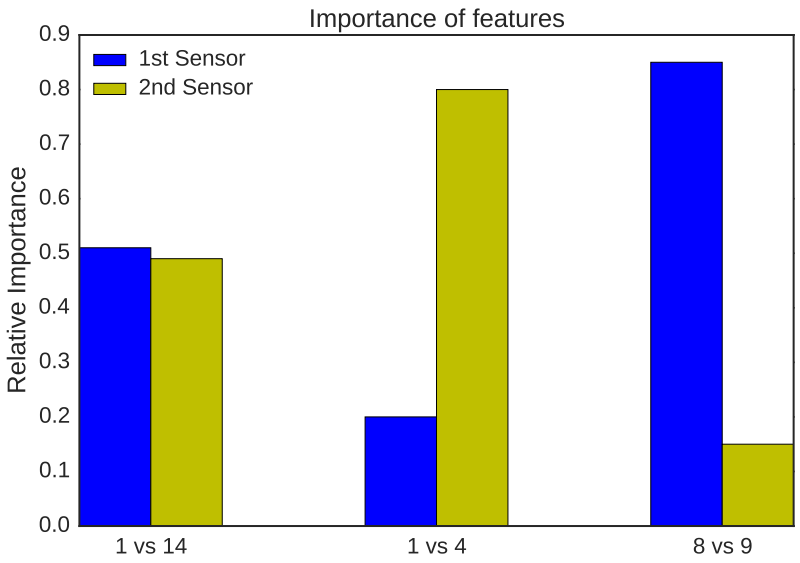


Figure 6.13: Feature importance of three tested combinations

Locations 1 and 4 are both close to the second sensor, so the second sensor determines most of the samples (about 80%), as shown in the figure. Location 4 is about 5 m from sensor 2 and is about 11 m from sensor 1. In addition, the distance from the attacker to the legitimate device is about 4 m. Locations 8 and 9 are close the first sensor, thus the first sensor determines which samples belong to which class for the majority of samples (about 85%), as shown in the figure. Location 8 is about 2 m away from the first sensor and about 10 m away from the second sensor. Location 9 is about 4 m away from the first sensor and 11 m away from the first sensor. The two locations are about 4 m away from each other.

6.3 Anomaly Detection Results and Discussion

The decision boundary that separates the legitimate device data points and the attacker is similar to the one that is shown in Figure 6.14 for most of the combinations. A hyperplane is drawn around the legitimate device to isolate it from the outliers as shown in the figure. The data point that falls inside the drawn decision boundary is classified as normal, and the data point that falls outside the decision boundary is classified as abnormal.

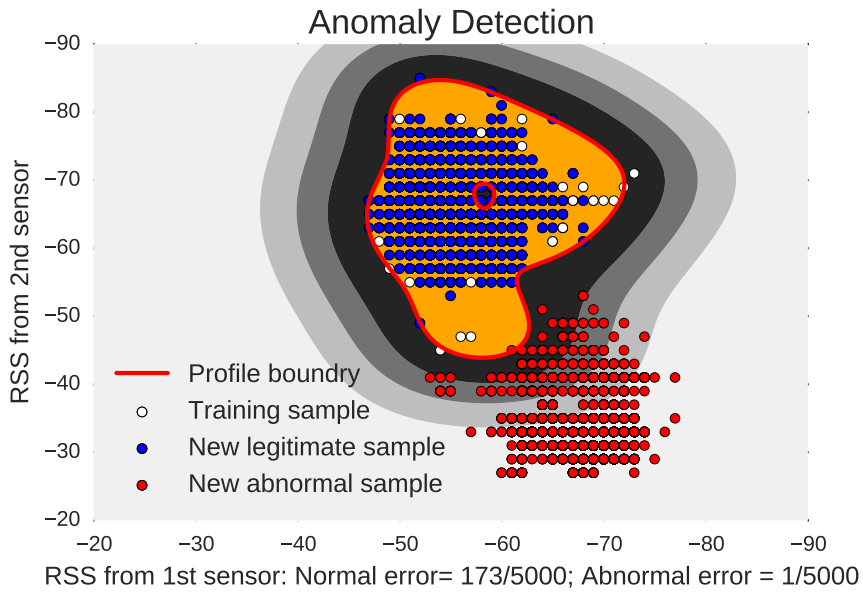


Figure 6.14: Anomaly detection decision boundary and data separation.

Two important parameters that control the decision boundary are γ and ν. We choose to make big decision boundary to minimize the false alerts. The accuracy of the misuse detection is better than the anomaly detection. The overall accuracy of the anomaly detection framework shown in Table 6.12 is 79.20% for all location combinations, 63.63% when the distance between the attacker and the legitimate device is less than 4 m, 72.85%

when the distance between the two devices is between 4 and 8 m, and 93.60% when the distance is between 8 and 13 m.

Table 6.12: Novelty detection accuracy

Mean By Distance	Normal Accuracy	Abnormal Accuracy	Total
Overall mean	98.12	60.28	79.20
4 m mean	98.44	28.83	63.63
8 m mean	98.15	47.55	72.85
8-13 m mean	97.94	89.27	93.60

However, the anomaly detection is suitable for situations where it is hard to cover the whole area (e.g., a company in a three floor building that has neighbouring companies, the anomaly detection can create a profile for the legitimate device). It can reduce the training overhead by only training the legitimate device samples instead of simulating the existence of the attacker in every possible location in the area. Also, the detection time is acceptable; the average detection time is only 8.3 ms as shown in Figure 6.15.

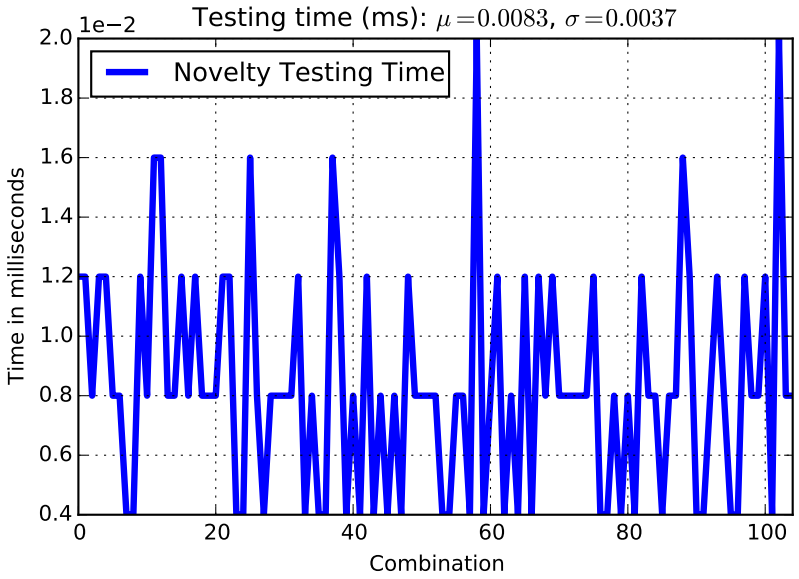


Figure 6.15: Anomaly detection testing time.

6.4 MAC Address Spoofing Detection by Majority Vote

We applied the majority voting technique to detect MAC address spoofing. The accuracy of the majority voting is slightly better than our approach that is based on Random Forests. The average accuracy shown in Table 6.13 is 94.83% and the standard deviation is so low in comparison to the previously mentioned approaches including our approach.

Table 6.13: Majority voting detection accuracy.

mean	std	min	50%	75%	max
94.83181	7.105315	71.35	98.81	99.92	100

The detection time is more expensive than the previous solutions including ours. However, the detection can be done in real-time, because the detection time per frame is still in microseconds. The average testing time shown in Figure 6.16 is about 192 ms.

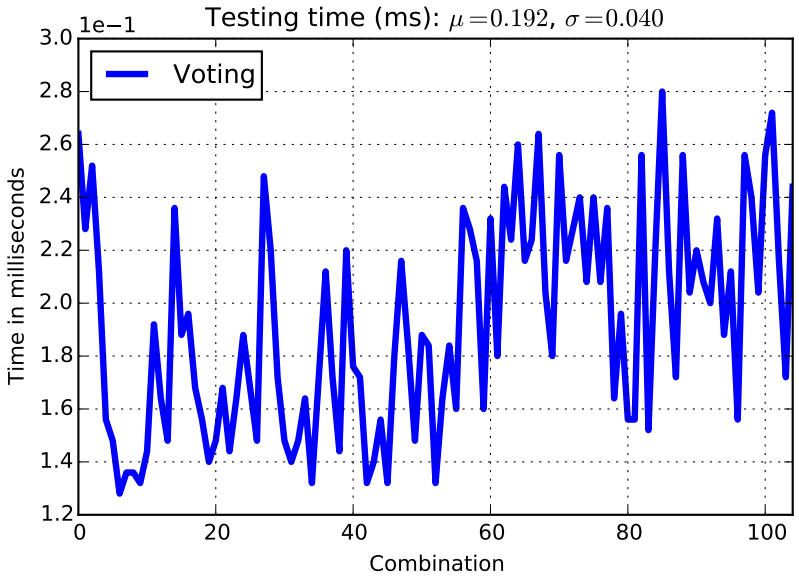
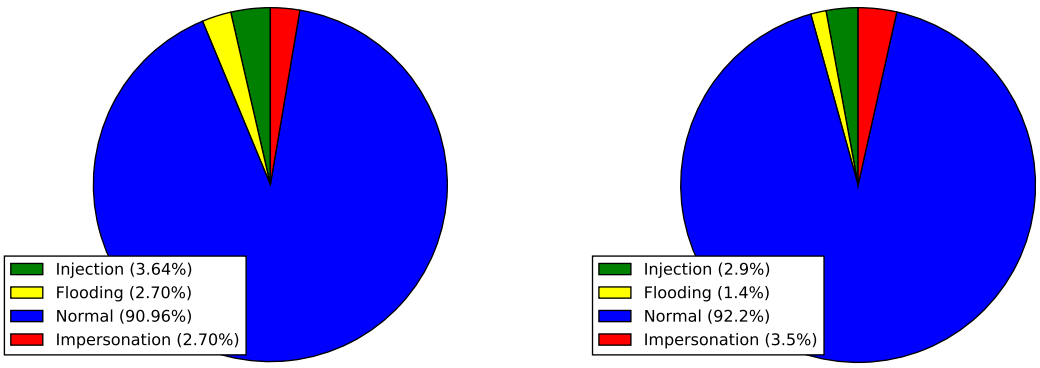


Figure 6.16: Majority Voting testing time.

6.5 Public Dataset Results and Discussion

The only public data-set that we know for WLANs is introduced in [3]. The data-set includes four parts, which are two reduced data-sets for researches interested in Wireless Intrusion Detection Systems (WIDSs) and two full data sets for big data researches. The two reduced data-sets consists of four classes and fifteen classes, respectively. The four classes are the categories that the launched attacks belong to (which are flooding, injection, and impersonation) and the normal class while the other reduced data-set consists of the names of the launched attacks and the normal class. The number of training samples of each reduced data-set is 1,795,575 and the number of testing samples is 575,643. Figure 6.17 shows the percentage of each class in both the training set and the testing set. The number of features is 156 features, representing the WLAN frame fields along with physical layer meta-data.



(a) (b)

Figure 6.17: The dataset records. (**a**) training set records; (**b**) testing set records

6.5.1 802.11 Attacks

The attacks that are launched by the authors (who published the data-set) were based on WEP, but most of the attacks share the same characteristics on the other security mechanisms. In this subsection we will explain the classes that are used in the reduced data-set and how the 20 features have been selected by the data mining technique.

Injection Attacks

Flood the wireless network with encrypted data frames of smaller size than the normal frames. ARP injection attack is an attack of which the attacker launches to speed up the process of collecting Initialization Vectors (IVs) from the targeted wireless device or AP in a small amount of time. Some penetration testing tools (such as Aireplay) are used to launch this attack and use the same *IV values*, which cannot occur under normal conditions. Also, the *DS status flag* is always set to 1 for all the frames sent during ARP injection.

Another vital attack is fragmentation, in which the attacker injects small fragmented data frames. This attack usually takes about a second if succeeded. Some of the penetration testing tools that launch this attack use a static invalid *Destination Address*; the *DS status flag* is always set to 1, *the frame length* is small but is not fixed, and the frames have out-of-order *sequence number*.

Flooding Attacks

usually generate an increase in the frames in the WLAN, the management frames in particular. However, it is not always a valid assumption to consider the increase of the management frames as indication of flooding attacks; sometimes it could be an indication of malfunction of certain device. Although the attacker can masquerade as a legitimate device, it is much harder to hide the increase of the management frames produced by flooding attacks. For example, a de-authentication attack is launched by some tools using the same *reason code* and has an out-of-order *sequence number*.

Also, some tools (such as MDK3 that the hackers use to launch authentication flooding and beacon flooding attacks) use a *sequence number* that is always set to 0. Tools such as Metaspolit (used to launch probe response flooding attack) use a random sender address, which could have a valid 24-bit number that identifies the vendor uniquely. This is known as Organizationally Unique Identifier (OUI).

Impersonation Attacks

masquerades as one of the legitimate devices in WLAN by changing one or more of its characteristics. Evil-twin AP is one example, where the attacker can change the MAC address and Service Set Identifier (SSID) of the device to be the same as the MAC address and SSID of the existing AP. Such attacks are always proceeded by de-authentication attacks targeting wireless devices that are connecting to the targeted AP, to force them to connect back to the fake AP. This attack is launched by tools like Airbase, which sends

broadcast beacon frames with fixed *frame length*. Furthermore, in all impersonation attacks, the *Received Signal Strength (RSS)* of the attacker is different than the legitimate device RSS if there is a significant distance between the two devices.

6.5.2 Data set Limitations

- It only applied on WEP encryption method, some of the features are WEP-dependent. The majority of the attacks in the data set can be applied on other security standards (such as WPA, WPA2 and 802.11w amendment), but some of them are WEP-specific.
- Most of the attacks are launched by specific penetration testing tools to build the patterns of the intrusions; attackers might use different existing or customized tools to exploit some of the wholes and bypass the IDS.
- Does not consider some cases, the mobility of the attacker in particular.

The best machine learning algorithms that we used in our experiments are Decision Trees, Extra Trees, and Random Forests. Decision Trees is not stable; we ran the test several times and it gave us different results every time. The three classifiers did not achieve better results than the J48 classifier that the authors of [3] used in their experiments. We decided to use Bagging classifier of minimum Decision Trees as a base estimator to be more robust and to have minimum time. Bagging classifier yields slightly better results and better timing. We then used the voting classifier that utilized Extra Trees of 20 trees, Random Forests of 20 trees, and Bagging classifier of 10 Decision Trees as base estimator and got better results and better time.

6.5.3 Bagging

We used Decision Tree [142] (introduced by Breiman) et al. as a base estimator to build the Bagging method. A number of 10 trees was used to minimize the cost. Table 6.14 shows the confusion matrix of the bagging method.

Table 6.14: Bagging

Normal	Flooding	Injection	Impersonation	Classified as
530383	343	0	59	Normal
2585	5512	0	0	Flooding
2	0	**16680**	0	Injection
18606	2	0	**1471**	Impersonation

Among the three tested classifiers, it is the most accurate classifier for the hardest class, which is the impersonation class. It is also slightly better than our voting classifier, of which about 1471 to 1470 occurrences classified correctly. Bagging and Extra Trees classifiers are better than the rest of the classifiers (including the voting technique) in classifying the injection class of 16680 occurrences (i.e., it misclassified only 2 occurrences). It is expensive in term of time (about 154 seconds) in comparison to Random Forests and Extra Trees ensemble methods. The overall accuracy of the bagging method is 96.25%, as shown in Table 6.15.

Table 6.15: All Features

Method	Accuracy	Time
Extra Trees	96.06	18.1
Random Forests	95.89	22.4
Bagging	96.25	154
Voting	96.32	390
Kolias et al. [3]	96.20	3921.68

The accuracy did not change when we used the reduced features, but the time has decreased of about 35.7 seconds as shown in Table 6.16.

Table 6.16: 20 Features

Method	Accuracy	Time
Extra Trees	96.31	8.03
Random Forests	96.31	9.95
Bagging	96.25	35.7
Voting	96.32	107
Kolias et al. [3]	96.26	568.92

6.5.4 Random Forests

We used 20 trees to build the ensemble because Random Forests is lighter than Bagging method. The accuracy of Random Forests is the worst among the tested methods when we used the entire feature set of about 95.89% (as shown in Table 6.15). However, it is the best method to classify flooding class. The training time is second after Extra Trees classifier of about 22.4 seconds when using all of the feature set and 9.95 seconds when using the reduced feature set. It is the algorithm that most likely benefited from reducing the feature set in term of accuracy. It jumped from 95.89% to 96.31% after we applied the feature selection technique. Table 6.17 shows the confusion matrix of Random Forests; it is the best method that classifies the flooding class correctly.

Table 6.17: Random Forests

Normal	Flooding	Injection	Impersonation	Classified as
530775	6	0	4	Normal
2536	**5561**	0	0	Flooding
41	0	16641	0	Injection
18645	0	0	1434	Impersonation

6.5.5 Extra Trees

We also used 20 trees to build the ensemble of Extra trees. The overall accuracy of Extra Trees is 96.06% when we used the whole feature set. It improved to 96.31% when we applied the feature selection capability. The best time among the tested algorithms is the time of Extra Trees (about 18.1 seconds) when using the whole feature set and 8.03 seconds when we applied the reduced feature set. Aside from the Bagging method, Extra Trees classified 16680 occurrences of injection class correctly (as shown in Table 6.18).

Table 6.18: Extra Trees

Normal	Flooding	Injection	Impersonation	Classified as
530773	2	0	10	Normal
2601	5496	0	0	Flooding
2	0	**16680**	0	Injection
18619	0	0	1460	Impersonation

6.5.6 Majority Voting

The Majority Voting relies on the base classifiers. We chose light classifiers to get better results and to be able to detect intrusions in real time. As expected, it is the best method in term of accuracy (about 96.32%) when using the whole feature set. The time is expensive, about 390 seconds. It is the best method to classify the normal class. As shown in Table 6.19, the method 100% correctly classified the normal occurrences as normal (i.e., there is no false positive at all). It also maintained its accuracy; the best method in term of accuracy when we reduced the feature set. The time decreased significantly when we reduce the feature set from about 390 seconds to 107 seconds, using the full feature set.

Table 6.19: Voting Classifier

Normal	Flooding	Injection	Impersonation	Classified as
530778	0	0	0	Normal
2589	5508	0	0	Flooding
5	0	16677	0	Injection
18609	0	0	1470	Impersonation

Figure 6.18 shows the overall details of the four classes, the correctly classified and mis-classified occurrences. The normal class has been classified 100% correctly, the flooding class classification rate is good, the error rate is about 32%, the injection class error rate is so low (only 0.03%), while the impersonation error rate is high because most of the attacks that belong to the impersonation class are in the testing set, but not in the training set.

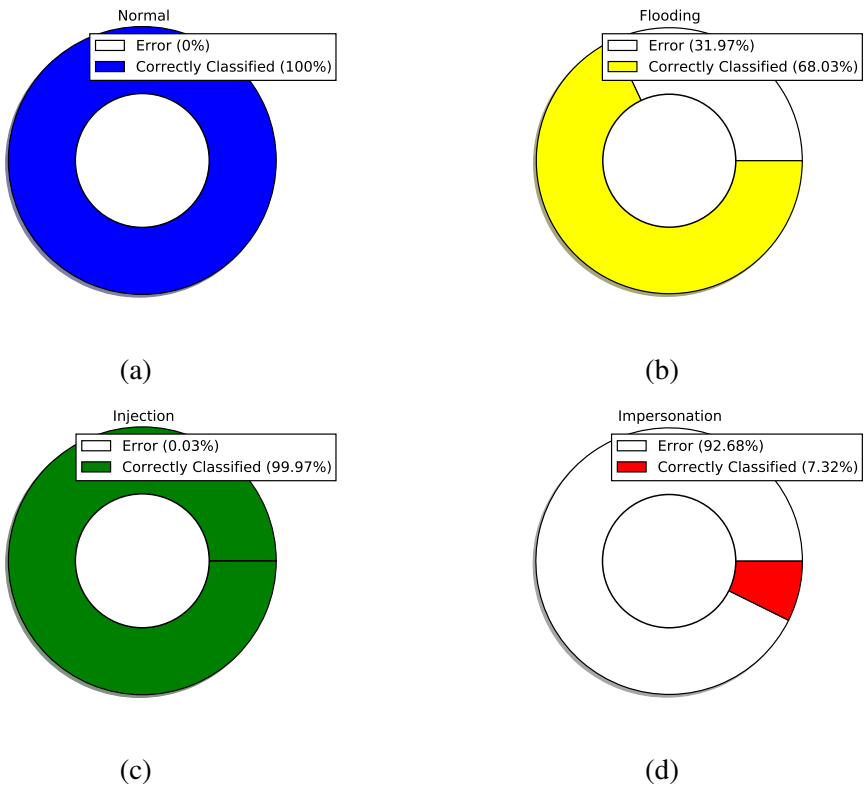


Figure 6.18: Each class classification accuracy. (**a**) normal class accuracy; (**b**) flooding class accuracy; (**c**) injection class accuracy; (**c**) impersonation class accuracy

The accuracy improvement was not significant. Our method accuracy is slightly better than Kholias et al.'s best performing algorithm (i.e., 96.32% to 96.19% when we used the entire feature set and 96.32% to 96.26% using the reduced feature set). However, the computation time has improved significantly; Kholias et al.'s best performing algorithm in term of accuracy takes about 3922 seconds using the entire feature set and 569 seconds using the reduced feature set. Our method takes only 390 seconds when using the entire feature set and 107 seconds when we reduced the feature set of 20 features.

6.5.7 Most Important 20 Features

Figure 6.19 shows the most important features selected by Extra Tree ensemble method.

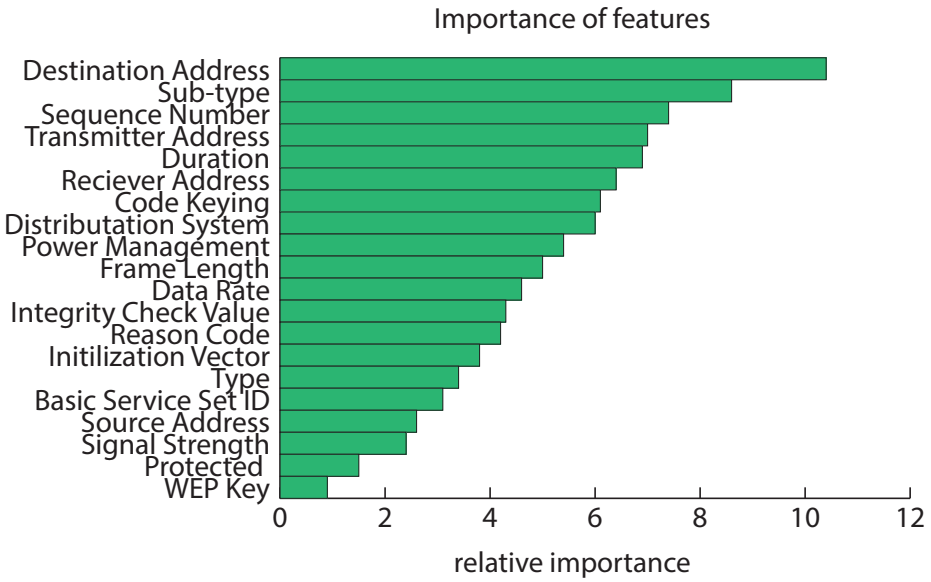


Figure 6.19: Most important 20 features.

The most important 20 features that have been selected are as follows:

- *Destination Address(DA)* is the final destination of the data frame.
- *Sub-type* is in the control frame which identifies the purpose of the frame type. For instance, if the type of the frame is control, the sub-type field could be one of the possible sub-types such as CTS, RTS, Ack and so on.
- *Seq:* every 802.11 frame has a sequence number except of control frames. The sequence number is incremented by one from 0 to 4,095 of every consecutive frame.

- *Transmitter Address(TA)* is one of two addresses that the frame might be transmitted from which are the first originator of the frame (i.e., the wireless users) or the intermediate address that transfer the frame to the final destination (i.e., the AP).

- *Duration* field identifies the time required to transmit the frame in microseconds.

- *Receiver Address (RA)* is the first device that receives the data frame, it could be the AP in the path to the final destination or the device that receives the frame which is the final destination.

- *Type.cck* (Complementary Code Keying) is a modulation scheme that is adopted to achieve high data rates.

- *fc.ds* is the distribution system status field that indicates which direction the frame is going to.

- *pwrmgt* indicates if the station is either going to change its status to power save mode or can receive frames.

- *frame-len* indicates the length of the frame in the wire.

- *datarate* specifies the supported data rate.

- *wep.icv* (Integrity Check Value) is a 4 byte long that is calculated using the frame and attached to it.

- *reason c* there are some reasons to be indicated when sending a deauthentication frame such as station is leaving or disassociated due to inactivity.

- *wep.iv* (WEP Initialization Vector) is a 24 bits long that is sent in the clear, different for each encrypted frame and concatenated with the fixed root key.
- *type* has to be one of data, control, or management.
- *bssid* is the MAC address of the AP.
- *Source Address (SA)* of the frame originator.
- *RSS* is the Received Signal Strength (RSS) of the sender measured at the receiver.
- *protected* indicates the encryption method that is used by the WLAN network.
- *wep.key* (Wired Equivalence Privacy) key that is a hexadecimal number that encrypts messages between group of connected devices in WLAN. There are two key sizes that WEP supports which are 40 bits and 104 bits.

CHAPTER 7: CONCLUSION

We proposed a technique based on Random Forests ensemble method which characterizes the shape of a dataset to detect MAC address spoofing, instead of assuming that the data are Gaussian-distributed. All previous methods based on clustering algorithms assume that there are two clusters, which is not a good assumption because one device, such as an AP, can form two clusters. Based on our extensive experiments and evaluations, we determined that our proposed method performs very well in terms of accuracy and prediction time. We proposed a technique to detect MAC address spoofing based on Random Forests, as it outperforms all the clustering algorithms-based approaches that were proposed previously, in terms of accuracy. Furthermore, it has a good prediction time. We also proposed an outlier or novelty detection method to detect MAC address spoofing. Outlier/novelty detection methods only require training using a legitimate device without covering the whole network range. We used an approach that is based on a one-class SVM to build a profile for legitimate devices.

Furthermore, we improved the accuracy and the time on the AWID data-set using a classifier that votes on the output of the carefully picked three classifiers (which are Extra Trees, Random Forests, and Bagging with ten Decision Trees as base estimators)

which performs well in both accuracy and time. The best performing classifier is the voting classifier which improved the accuracy and the time to 96.31% and 390 seconds when we used all the features. We also used a data mining technique based on Extra Trees ensemble method to choose the best 20 features to decrease time and improve accuracy of the best performing classifiers. We maintain the same accuracy, but improved the time of about 107 seconds.

In this research we assumed the mobility of the attacker to detect MAC address spoofing, but the legitimate device should be static for the detection to be succeeded. In the future, we will consider the mobility of both the legitimate device and the spoofing device. We would also investigate location determination in both WLANs and WSNs after spoofing detection.

REFERENCES

[1] M.-W. Park, Y.-H. Choi, J.-H. Eom, and T.-M. Chung, "Dangerous wi-fi access point: Attacks to benign smartphone applications," *Personal and Ubiquitous Computing*, vol. 18, no. 6, pp. 1373–1386, 2014.

[2] R. C. Carrano, L. Magalhaes, D. C. M. Saade, and C. V. Albuquerque, "Ieee 802.11 s multihop mac: A tutorial," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 1, pp. 52–67, 2011.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[4] H. Han, B. Sheng, C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue ap detection," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 11, pp. 1912–1925, Nov. 2011.

[5] B. Alotaibi and K. Elleithy, "A passive fingerprint technique to detect fake access points," in *Wireless telecommunications symposium (WTS)*, IEEE, 2015, pp. 1–8.

[6] L. Ma, A. Y. Teymorian, and X. Cheng, "A hybrid rogue access point protection framework for commodity wi-fi networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, IEEE, 2008.

[7] W. Wei, K. Suh, B. Wang, *et al.*, "Passive online rogue access point detection using sequential hypothesis testing with tcp ack-pairs," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ACM, 2007, pp. 365–378.

[8] H. Yin, G. Chen, and J. Wang, "Detecting protected layer-3 rogue aps," in *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, IEEE, 2007, pp. 449–458.

[9] B. Alotaibi and K. Elleithy, "An empirical fingerprint framework to detect rogue access points," in *Systems, applications and technology conference (LISAT), 2015 IEEE Long Island*, IEEE, 2015, pp. 1–7.

[10] S. Shetty, M. Song, and L. Ma, "Rogue access point detection by analyzing network traffic characteristics," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, IEEE, 2007, pp. 1–7.

[11] B. Alotaibi and K. Elleithy, "Rogue access point detection: Taxonomy, challenges, and future directions," *Wireless Personal Communications*, vol. 90, no. 3, pp. 1261–1290, 2016.

[12] N. Agrawal and S. Tapaswi, "Wireless rogue access point detection using shadow honeynet," *Wireless Personal Communications*, vol. 83, no. 1, pp. 551–570, 2015.

[13] R. Beyah and A. Venkataraman, "Rogue-access-point detection: Challenges, solutions, and future directions," *IEEE Security & Privacy*, vol. 9, no. 5, pp. 56–61, 2011.

[14] G. Shivaraj, M. Song, and S. Shetty, "A hidden markov model based approach to detect rogue access points," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, IEEE, 2008, pp. 1–7.

[15] A.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. W. Hong, "A flow-based method for abnormal network traffic detection," in *Network operations and management symposium, 2004. NOMS 2004. IEEE/IFIP*, IEEE, vol. 1, 2004, pp. 599–612.

[16] Y. Chen, W. Trappe, and R. Martin, "Detecting and localizing wireless spoofing attacks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, IEEE, Jun. 2007, pp. 193–202.

[17] Y. Chen, J. Yang, W. Trappe, and R. P. Martin, "Detecting and localizing identity-based attacks in wireless and sensor networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 5, pp. 2418–2434, 2010.

[18] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Undesired relatives: Protection mechanisms against the evil twin attack in ieee 802.11," in *Proceedings of the 10th acm symposium on qos and security for wireless and mobile networks*, ACM, 2014, pp. 87–94.

[19] C. Yang, Y. Song, and G. Gu, "Active user-side evil twin access point detection using statistical techniques," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 5, pp. 1638–1651, 2012.

[20] H. Mustafa and W. Xu, "Cetad: Detecting evil twin access point attacks in wireless hotspots," in *Communications and Network Security (CNS), 2014 IEEE Conference on*, IEEE, 2014, pp. 238–246.

[21] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proceedings of the 5th ACM workshop on Wireless security*, ACM, 2006, pp. 43–52.

[22] A. Ordi, H. Mousavi, B. Shanmugam, M. R. Abbasy, and M. R. N. Torkaman, "A novel proof of work model based on pattern matching to prevent dos attack," in *Digital Information and Communication Technology and Its Applications*, Springer, 2011, pp. 508–520.

[23] A. Ordi, M. Zamani, N. B. Idris, A. A. Manaf, and M. S. Abdullah, "A novel wlan client puzzle against dos attack based on pattern matching," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–12, 2015.

[24] E. Kartsakli, A. S. Lalos, A. Antonopoulos, *et al.*, "A survey on m2m systems for mhealth: A wireless communications perspective," *Sensors*, vol. 14, no. 10, pp. 18 009–18 052, 2014.

[25] J. Serra, D. Pubill, A. Antonopoulos, and C. Verikoukis, "Smart hvac control in iot: Energy consumption minimization with user comfort constraints," *The Scientific World Journal*, vol. 2014, pp. 1–11, 2014.

[26] E. M. Shakshuki, N. Kang, and T. R. Sheltami, "Eaack—a secure intrusion detection system for manets," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 3, pp. 1089–1098, 2013.

[27] X. Li, J. Ma, and Y. Shen, "An efficient wlan initial authentication protocol," in *Proc. IEEE Global Comm. Conf.(Globecom'12)*, IEEE, 2012.

[28] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13–21, 2015.

[29] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[30] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 5, pp. 649–659, 2008.

[32] D. S. Kim, S. M. Lee, and J. S. Park, "Building lightweight intrusion detection system based on random forest," in *Advances in Neural Networks-ISNN 2006*, Springer, 2006, pp. 224–230.

[33] D. DeBarr and H. Wechsler, "Spam detection using clustering, random forests, and active learning," in *Sixth Conference on Email and Anti-Spam. Mountain View, California*, Citeseer, 2009.

[34] A. A. Akinyelu and A. O. Adewumi, "Classification of phishing email using random forest machine learning technique," *Journal of Applied Mathematics*, vol. 2014, pp. 1–6, 2014.

[35] M. Tahir, N. Javaid, A. Iqbal, Z. A. Khan, and N. Alrajeh, "On adaptive energy-efficient transmission in wsns," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–10, 2013.

[36] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 mac layer spoofing using received signal strength," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, IEEE, 2008.

[37] H. Alipour, Y. B. Al-Nashif, P. Satam, and S. Hariri, "Wireless anomaly detection based on ieee 802.11 behavior analysis," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 10, pp. 2158–2170, 2015.

[38] A. Alabdulatif, X. Ma, and L. Nolle, "Analysing and attacking the 4-way handshake of ieee 802.11 i standard," in *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, IEEE, 2013, pp. 382–387.

[39] R. Singh and T. P. Sharma, "On the ieee 802.11 i security: A denial-of-service perspective," *Security and Communication Networks*, vol. 8, no. 7, pp. 1378–1407, 2015.

[40] T. D. Nguyen, D. H. Nguyen, B. N. Tran, H. Vu, and N. Mittal, "A lightweight solution for defending against deauthentication/disassociation attacks on 802.11 networks," in *Computer Communications and Networks, 2008. ICCCN'08. Proceedings of 17th International Conference on*, IEEE, 2008, pp. 1–6.

[41] M. Agarwal, S. Biswas, and S. Nandi, "Detection of deauthentication denial of service attack in 802.11 networks," in *India Conference (INDICON), 2013 Annual IEEE*, IEEE, 2013, pp. 1–6.

[42] K. Tao, J. Li, and S. Sampalli, "Detection of spoofed mac addresses in 802.11 wireless networks," in *E-business and Telecommunications*, Springer, 2007, pp. 201–213.

[43] F. Guo and T.-c. Chiueh, "Sequence number-based mac address spoof detection," in *Recent Advances in Intrusion Detection*, Springer, 2005, pp. 309–329.

[44] J. Mar, Y.-C. Yeh, and I.-F. Hsiao, "An anfis-ids against deauthentication dos attacks for a wlan," in *Information Theory and its Applications (ISITA), 2010 International Symposium on*, IEEE, 2010, pp. 548–553.

[45] J. Yang, Y. Chen, W. Trappe, and J. Cheng, "Determining the number of attackers and localizing multiple adversaries in wireless spoofing attacks," in *INFOCOM 2009, IEEE*, IEEE, 2009, pp. 666–674.

[46] J. Yang, W. Trappe, Y. Chen, and J. Cheng, "Detection and localization of multiple spoofing attackers in wireless networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 44–58, 2013.

[47] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, and D. S. Wallach, "Robotics-based location sensing using wireless ethernet," *Wireless Networks*, vol. 11, no. 1-2, pp. 189–204, 2005.

[48] K. Bicakci and B. Tavli, "Denial-of-service attacks and countermeasures in ieee 802.11 wireless networks," *Computer Standards & Interfaces*, vol. 31, no. 5, pp. 931–941, 2009.

[49] S. Willens, A. C. Rubens, C. Rigney, and W. A. Simpson, "Remote authentication dial in user service (radius)" 2000.

[50] M. El Rifai and P. K. Verma, "An ieee 802.11 quantum handshake using the three-stage protocol," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, IEEE, 2014, pp. 1–6.

[51] H. Alipour, Y. Al-Nashif, and S. Hariri, "Ieee 802.11 anomaly-based behavior analysis," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*, Jan. 2013, pp. 369–373.

[52] "Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 4: Protected management frames," *IEEE Std 802.11w-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, and IEEE Std 802.11y-2008)*, pp. 1–111, Sep. 2009.

[53] V. Roth, W. Polak, E. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *Proceedings of the first ACM conference on Wireless network security*, ACM, 2008, pp. 220–235.

[54] Y. Song, C. Yang, and G. Gu, "Who is peeping at your passwords at starbucks? – to catch an evil twin access point," in *2010 IEEE/IFIP International Conference on Dependable Systems&Networks (DSN)*, IEEE, 2010, pp. 323–332.

[55] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A measurement based rogue ap detection scheme," in *INFOCOM 2009, IEEE*, IEEE, 2009, pp. 1593–1601.

[56] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, 2006, pp. 581–590.

[57] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor, "Crying wolf: An empirical study of ssl warning effectiveness.," in *USENIX Security Symposium*, 2009, pp. 399–416.

[58] L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, "Rap: Protecting commodity wi-fi networks from rogue access points," in *The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness & Workshops*, ACM, 2007, p. 21.

[59] K. K. Raju, V. Vallikumari, and K. Raju, "Modeling and analysis of ieee 802.11 i wpa-psk authentication protocol," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, IEEE, vol. 5, 2011, pp. 72–76.

[60] X. Li, F. Bao, S. Li, and J. Ma, "Flap: An efficient wlan initial access authentication protocol," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 2, pp. 488–497, 2014.

[61] C. He, "Analysis of security protocols for wireless networks," PhD thesis, Stanford University, 2005.

[62] Y. Wang, Z. Jin, and X. Zhao, "Practical defense against wep and wpa-psk attack for wlan," in *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, IEEE, 2010, pp. 1–4.

[63] S. Onno, R. Gelloz, O. Heen, and C. Neumann, "User-based authentication for wireless home networks," in *Consumer Electronics-Berlin (ICCE-Berlin), 2012 IEEE International Conference on*, IEEE, 2012, pp. 218–220.

[64] "Ieee standard for local and metropolitan area networks - port-based network access control," *IEEE Std 802.1X-2004 (Revision of IEEE Std 802.1X-2001)*, pp. 1–175, Dec. 2004.

[65] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, *et al.*, *Extensible authentication protocol (eap)*, 2004.

[66] P. Robyns, B. Bonné, P. Quax, and W. Lamotte, "Short paper: Exploiting wpa2-enterprise vendor implementation weaknesses through challenge response oracles," in *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*, ACM, 2014, pp. 189–194.

[67] C.-I. Fan, Y.-H. Lin, and R.-H. Hsu, "Complete eap method: User efficient and forward secure authentication protocol for ieee 802.11 wireless lans," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 672–680, 2013.

[68] K. M. Ali and A. Al-Khlifa, "A comparative study of authentication methods for wi-fi networks," in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2011 Third International Conference on*, IEEE, 2011, pp. 190–194.

[69] M. M. Marlinspike, D. Hulton, and M. Ray, "Defeating pptp vpns and wpa2 enterprise with ms-chapv2," *Defcon, July* 2012.

[70] N. Asokan, V. Niemi, and K. Nyberg, "Man-in-the-middle in tunnelled authentication protocols," in *Security Protocols*, Springer, 2003, pp. 28–41.

[71] A. Hassan and X. Zhang, "Bypassing web-based wireless authentication systems," in *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island*, IEEE, 2011, pp. 1–4.

[72] R. M. Pandurang and D. C. Karia, "Performance measurement of wep and wpa2 on wlan using openvpn," in *Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on*, IEEE, 2015, pp. 1–4.

[73] C. Neumann, O. Heen, and S. Onno, "An empirical study of passive 802.11 device fingerprinting," in *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, IEEE, 2012, pp. 593–602.

[74] A. S. Uluagac, S. V. Radhakrishnan, C. Corbett, A. Baca, and R. Beyah, "A passive technique for fingerprinting wireless devices with wired-side observations," in *Communications and Network Security (CNS), 2013 IEEE Conference on*, IEEE, 2013, pp. 305–313.

[75] D. Mónica and C. Ribeiro, "Wifihop-mitigating the evil twin attack through multihop detection," in *Computer Security–ESORICS 2011*, Springer, 2011, pp. 21–39.

[76] B. Sieka, "Active fingerprinting of 802.11 devices by timing analysis," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, IEEE, vol. 1, 2006, pp. 15–19.

[77] T. Kim, H. Park, H. Jung, and H. Lee, "Online detection of fake access points using received signal strengths," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, IEEE, 2012, pp. 1–5.

[78] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles, "Active behavioral fingerprinting of wireless devices," in *Proceedings of the first ACM conference on Wireless network security*, ACM, 2008, pp. 56–61.

[79] S. Nikbakhsh, A. B. A. Manaf, M. Zamani, and M. Janbeglou, "A novel approach for rogue access point detection on the client-side," in *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on*, IEEE, 2012, pp. 684–687.

[80] S. Chae, H. Jung, I. Bae, and K. Jeong, "A scheme of detection and prevention rogue ap using comparison security condition of ap," in *Advances in Computer Science and Electronics Engineering, 2012 Universal Association of Computer and Electronics Engineers International Conference on*, 2012, pp. 302–306.

[81] G. Qu and M. M. Nefcy, "Rapid: An indirect rogue access points detection system," in *Performance Computing and Communications Conference (IPCCC), 2010 IEEE 29th International*, IEEE, 2010, pp. 9–16.

[82] K. F. Kao, W. C. Chen, J. C. Chang, and H. Te Chu, "An accurate fake access point detection method based on deviation of beacon time interval," in *Software Security and Reliability-Companion (SERE-C), 2014 IEEE Eighth International Conference on*, IEEE, 2014, pp. 1–2.

[83] P. Chumchu, T. Saelim, and C. Sriklauy, "A new mac address spoofing detection algorithm using plcp header," in *Information Networking (ICOIN), 2011 International Conference on*, IEEE, 2011, pp. 48–53.

[84] C. Szongott, M. Brenner, and M. Smith, "Metds-a self contained, context-based detection system for evil twin access points," in *Financial Cryptography and Data Security*, Springer, 2015, pp. 370–386.

[85] T. Cross and T. Takahashi, "Secure open wireless access," in *Black Hat–USA 2011*, 2011.

[86] K. Bauer, H. Gonzales, and D. McCoy, "Mitigating evil twin attacks in 802.11," in *Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International*, IEEE, 2008, pp. 513–516.

[87] H. Gonzales, K. Bauer, J. Lindqvist, D. McCoy, and D. Sicker, "Practical defenses for evil twin attacks in 802.11," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, IEEE, 2010, pp. 1–6.

[88] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the reliability of wireless fingerprinting using clock skews," in *Proceedings of the third ACM conference on Wireless network security*, ACM, 2010, pp. 169–174.

[89] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 3, pp. 449–462, 2010.

[90] F. Lanze, A. Panchenko, B. Braatz, and T. Engel, "Letting the puss in boots sweat: Detecting fake access points using dependency of clock skews on temperature," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, ACM, 2014, pp. 3–14.

[91] F. Lanze, A. Panchenko, I. Ponce-Alcaide, and T. Engel, "Hackers toolbox: Detecting software-based 802.11 evil twin access points," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, IEEE, 2015, pp. 225–232.

[92] W. Wei, S. Jaiswal, J. F. Kurose, and D. F. Towsley, "Identifying 802.11 traffic from passive measurements using iterative bayesian inference.," in *IEEE International Conference on Computer Communications (INFOCOM)*, IEEE, 2006.

[93] W. Wei, S. Jaiswal, J. Kurose, *et al.*, "Identifying 802.11 traffic from passive measurements using iterative bayesian inference," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 2, pp. 325–338, 2012.

[94] B. Yan, G. Chen, J. Wang, and H. Yin, "Robust detection of unauthorized wireless access points," *Mobile Networks and Applications*, vol. 14, no. 4, pp. 508–522, 2009.

[95] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, IEEE, vol. 4, 2004, pp. 2271–2275.

[96] C. D. Mano, A. Blaich, Q. Liao, *et al.*, "Ripps: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning," *ACM Transactions on Information and System Security (TISSEC)*, vol. 11, no. 2, p. 2, 2008.

[97] L. Watkins, R. Beyah, and C. Corbett, "A passive approach to rogue access point detection," in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, IEEE, 2007, pp. 355–360.

[98] M. K. Chirumamilla and B. Ramamurthy, "Agent based intrusion detection and response system for wireless lans," in *Communications, 2003. ICC'03. IEEE International Conference on*, IEEE, vol. 1, 2003, pp. 492–496.

[99] D. B. Faria and D. R. Cheriton, "Dos and authentication in wireless public access networks," in *Proceedings of the 1st ACM workshop on Wireless security*, ACM, 2002, pp. 47–56.

[100] B. Aslam, M. Akhlaq, and S. A. Khan, "802.11 disassociation dos attack simulation using verilog," *WSEAS Transactions on Communications*, vol. 7, pp. 198–206, 2008.

[101] C. He and J. C. Mitchell, "Analysis of the 802.11 i 4-way handshake," in *Proceedings of the 3rd ACM workshop on Wireless security*, ACM, 2004, pp. 43–50.

[102] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications: Amendment 6: Medium access control (mac) security enhancements," *IEEE Std 802.11i-2004*, pp. 1–190, Jul. 2004.

[103] A. Lockhart, *Deauthentication frame dos*, 2005.

[104] L. Wang and B. Srinivasan, "Analysis and improvements over dos attacks against ieee 802.11 i standard," in *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, IEEE, vol. 2, 2010, pp. 109–113.

[105] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions.," in *USENIX security*, 2003, pp. 15–28.

[106] H. Xia and J. Brustoloni, "Detecting and blocking unauthorized access in wi-fi networks," in *Networking 2004*, Springer, 2004, pp. 795–806.

[107] F. Anjum, S. Das, P. Gopalakrishnan, L. Kant, and B. Kim, "Security in an insecure wlan network," in *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, IEEE, vol. 1, 2005, pp. 292–297.

[108] J. Wright, "Detecting wireless lan mac address spoofing," *White Paper, January* 2003.

[109] D. Madory, "New methods of spoof detection in 802.11 b wireless networking," PhD thesis, Citeseer, 2006.

[110] M. Agarwal, D. Pasumarthi, S. Biswas, and S. Nandi, "Machine learning approach for detection of flooding dos attacks in 802.11 networks and attacker localization," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1035–1051, 2014.

[111] B. Alotaibi and K. Elleithy, "A new mac address spoofing detection technique based on random forests," *Sensors*, vol. 16, no. 3, p. 281, 2016.

[112] S. Vanjale and P. Mane, "A novel approach for elimination of rogue access point in wireless network," in *India Conference (INDICON), 2014 Annual IEEE*, IEEE, 2014, pp. 1–4.

[113] V. S. Sriram, G. Sahoo, and K. K. Agrawal, "Detecting and eliminating rogue access points in ieee-802.11 wlan-a multi-agent sourcing methodology," in *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, IEEE, 2010, pp. 256–260.

[114] J. W. Branch, N. L. Petroni Jr, L. Van Doorn, and D. Safford, "Autonomic 802.11 wireless lan security auditing," *IEEE Security & Privacy*, no. 3, pp. 56–65, 2004.

[115] Airmagnet, "Best practices for securing your wireless lan," *White paper* 2004.

[116] Airdefence, *Tired of rogues? solutions for detecting and eliminating rogue wireless networks, white paper*.

[117] Netstumbler. [Online]. Available: `www.netstumbler.com`.

[118] Wavelink. [Online]. Available: `www.wavelink.com`.

[119] Airdefense. [Online]. Available: `www.airdefense.net`.

[120] P. Bahl, J. Padhye, L. Ravindranath, *et al.*, "Dair: A framework for managing enterprise wireless networks using desktop infrastructure," HotNets, 2005.

[121] Airwave. [Online]. Available: www.airwave.com.

[122] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden naive bayes multiclass classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13 492–13 500, 2012.

[123] A. Jamdagni, Z. Tan, X. He, P. Nanda, and R. P. Liu, "Repids: A multi tier real-time payload-based intrusion detection system," *Computer Networks*, vol. 57, no. 3, pp. 811–824, 2013.

[124] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," *Procedia Engineering*, vol. 30, pp. 1–9, 2012.

[125] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (sso)," *Applied Soft Computing*, vol. 12, no. 9, pp. 3014–3022, 2012.

[126] B. Mager, P. Lundrigan, and N. Patwari, "Fingerprint-based device-free localization performance in changing environments," *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 11, pp. 2429–2438, 2015.

[127] X. Chen, A. Edelstein, Y. Li, *et al.*, "Sequential monte carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, IEEE, 2011, pp. 342–353.

[128] J. Wilson and N. Patwari, "A fade-level skew-laplace signal strength model for device-free localization with wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 6, pp. 947–958, 2012.

[129] C. Xu, B. Firner, R. S. Moore, *et al.*, "Scpl: Indoor device-free multi-subject counting and localization using radio signal strength," in *Information Processing in Sen-*

sor Networks (IPSN), 2013 ACM/IEEE International Conference on, IEEE, 2013, pp. 79–90.

[130] X. Li, J. Wang, and C. Liu, "A bluetooth/pdr integration algorithm for an indoor positioning system," *Sensors*, vol. 15, no. 10, pp. 24 862–24 885, 2015.

[131] S. Tennina, M. Di Renzo, E. Kartsakli, *et al.*, "Wsn4qol: A wsn-oriented healthcare system architecture," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[132] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[133] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: Data mining, inference and prediction," *Springer*, pp. 587–589, 2009.

[134] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[135] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[136] Z.-H. Zhou, *Ensemble methods: Foundations and algorithms*. CRC Press, 2012.

[137] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

[138] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of network and computer applications*, vol. 30, no. 1, pp. 114–132, 2007.

[139] A. Zainal, M. A. Maarof, S. M. Shamsuddin, *et al.*, "Ensemble classifiers for network intrusion detection system," *Journal of Information Assurance and Security*, vol. 4, no. 3, pp. 217–225, 2009.

[140] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4–20, 2008.

[141] B. Alotaibi and K. Elleithy, "A majority voting technique for wireless intrusion detection systems," in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, IEEE, Apr. 2016, pp. 1–6.

[142] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

APPENDIX A: LIST OF PUBLICATIONS

Journal Publications

1. B. Alotaibi and K. Elleithy, "A new mac address spoofing detection technique based on random forests," *Sensors*, vol. 16, no. 3, p. 281, 2016, IF - 2.245.

2. B. Alotaibi and K. Elleithy, "Rogue access point detection: Taxonomy, challenges,and future directions," *Wireless Personal Communications*, vol. 90, no. 3, pp. 1261–1290, 2016, IF - 0.701.

Conference Publications

1. B. Alotaibi and K. Elleithy, "A passive fingerprint technique to detect fake access points," in *Wireless telecommunications symposium (WTS)*, IEEE, 2015, pp. 1–8.

2. B. Alotaibi and K. Elleithy, "An empirical fingerprint framework to detect rogue access points," in *Systems, applications and technology conference (LISAT), 2015 IEEE Long Island*, IEEE, 2015, pp. 1–7.

3. B. Alotaibi and K. Elleithy, "A majority voting technique for wireless intrusion detection systems," in *2016 IEEE Long Island Systems, Applications and Technology*

Conference (LISAT), IEEE, Apr. 2016, pp. 1–6.

Posters

1. B. Alotaibi and K. Elleithy, "A Majority Voting Technique for Wireless Intrusion Detection Systems." *Faculty Research Day (FRD), University of Bridgeport, Bridgeport, CT*, 2016.

2. B. Alotaibi and K. Elleithy, "Misuse Wireless Intrusion Detection System Based on Voting Technique." *American Society for Engineering Education (ASEE) Zone Conference Proceedings, Kingston, RI*, 2016.